AFRL-ML-WP-TR-1999-4018

LIFE PREDICTION OF SEALS

BRICE N. CASSENTI
JOHN P. WESSON

UNITED TECHNOLOGIES CORPORATION
UNITED TECHNOLOGIES RESEARCH CENTER
411 SILVER LANE
EAST HARTFORD, CT 06108

NOVEMBER 1998

FINAL REPORT FOR 09/16/1996 – 02/27/1998

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

MATERIALS AND MANUFACUTURING DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-7750

NOTICE

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

William M. Warner

WILLIAM M. WARNER, Project Engineer
Nonstructural Materials Branch
Nonmetallic Materials Division

STEPHEN L. SZARUGA, Acting Chief
Nonstructural Materials Branch
Nonmetallic Materials Division

ROGER D. GRISWOLD, Assistant Chief
Nonmetallic Materials Division
Materials & Manufacturing Directoratae

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
NOVEMBER 1998

3. REPORT TYPE AND DATES COVERED
FINAL REPORT FOR 09/16/1996 - 02/27/1998

4. TITLE AND SUBTITLE
LIFE PREDICTION OF SEALS

5. FUNDING NUMBERS
C F33615-96-C-5062
PE 62102
PR 4347
TA 61
WU 07

6. AUTHOR(S)
BRICE N. CASSENTI
JOHN P. WESSON

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNITED TECHNOLOGIES CORPORATION
UNITED TECHNOLOGIES RESEARCH CENTER
411 SILVER LANE
EAST HARTFORD, CT 06108

8. PERFORMING ORGANIZATION REPORT NUMBER
R98-5.916.0003-15

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
MATERIALS AND MANUFACTURING DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB, OH 45433-7750
POC: DR. W.M. WARNER, AFRL.MLBT, 937-255-9016

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-ML-WP-TR-1999-4018

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Air Force Research Laboratory requires a method for predicting the relative life of O-rings produced by different manufacturers. O-rings manufactured to the same MIL-spec from different manufacturers usually use different material formulations. Differences in formulations can result in different properties as the O-rings age. The required test methods would be based on benchtop procedures capable of determining basic properties of new and aged seals, which affect sealing properties.

Two methods potentially capable of measuring the storage and loss moduli of O-rings under dynamic conditions were proposed. One method was based on measuring multiple rebounds of a pendulum dropped against the O-ring. The second was based on determining the force displacement response of O-rings squeezed between two parallel plates and excited over a series of frequencies ranging from 1 to 100 Hz. Test instruments were built to explore both methods. Silicone and nitrile O-rings from multiple manufacturers were aged under different temperature and pressure conditions for 168 and 504 hours to provide the test samples.

Computational methods were developed to analyze the data from these test methods. Two computer code versions were developed. The first was designed to run on a PC, and predicted the deformation and forces for O-rings for a variety of loading conditions. The second was a workstation version that automatically ran either of two finite element codes (MARC or ANSYS). The workstation versions have much more capability than the PC versions, but of course are more difficult to run.

over

14. SUBJECT TERMS
Seals, O-rings, Elastomers, "Life Prediction", Modeling, "Dynamic Testing"

15. NUMBER OF PAGES
142

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
SAR

13. ABSTRACT (Cont'd)

The test methods both proved capable of measuring the basic properties of the O-rings, with somewhat greater accuracy from the pendulum rebound method. Variations between different materials were determined. However, the aging method did not induce significant changes in each family of O-rings, preventing development of a true life prediction method.

The rebound test method has been retained and will be used in the future to determine fundamental properties of elastomer samples of non-standard geometries.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

Fluid leakage due to failed seals in military aircraft fuel and hydraulic systems can drastically effect equipment performance, component life cycle and military readiness. Equally important is the environmental damage caused by such leakage. A computational model to accurately assess the service life of various elastomeric seals in their various operational environments would have a profound impact on solving these problems.

Seal materials made by different manufacturers to the same specification do not necessarily use identical materials or processes. The requirement is that seals meet certain performance requirements. Under operating conditions changes, degradation and permanent deformation of the seal materials can and will occur. Operating conditions usually involve exposure to heat, pressure, vibration and operating fluids while the seals are compressed in a housing (gland). During exposure, relaxation and deformation of the seal can occur. The fundamental chemistry of the seal may change due to aging or operating fluid being absorbed into the seal material. All of these effects will change the performance characteristics of the seal. These changes will affect sealing performance.

Different materials and seal formulations are expected to respond differently under these aging conditions. The result is that the performance of seals from different manufacturers can vary significantly over time. A test method capable of distinguishing between seals made to the same specification would be of great value when choosing seal materials and manufacturers.
This program selected fluorosilicone and nitrile seals from three sources, and exposed them to artificial aging conditions up to 504 hours. After exposure, changes in weight and shape were measured and the mechanical behavior of these test seals was measured using dynamic methods. The methods involved the measurement of the decay of multiple rebounds by a pendulum striking test O-rings and the measurement of the force displacement-response of O-ring cyclically oscillated while compressed between two parallel plates.

Analysis of the rebound tests allowed determination of the storage and loss moduli of each O-ring material and the fundamental frequency associated with the first three bounces. Using this method, the basic material properties of new and aged O-rings could be determined and compared.

In a similar manner, during the parallel plate measurements, force-displacement curves were generated at discrete excitation frequencies and the results analyzed to determine storage and loss moduli for each sample at these discrete frequencies. Using this method, the basic material properties of new and aged O-rings were determined and compared.

Two forms were developed for the life prediction system: a personal computer (PC) version and a workstation version. The PC version runs in two modes: (1) a Visual Basic mode where the user is prompted for the inputs and (2) a DOS mode where the user must use DOS commands to complete a run. The DOS mode provides more flexibility but is considerably more difficult to use. The workstation versions include a modeling system based on the MARC finite element code and one based on the ANSYS finite element code. The ANSYS version currently lacks the versatility of the MARC version, but can readily simulate tests on O-ring samples. Each of the

four versions can simulate load histories and geometries to varying degrees of complexity, and can predict the history during a typical loading cycle. This can be combined with the long term properties to aid in predicting the O-ring life.

Three basic tests were examined for obtaining the properties of the life prediction system. These included: a pendulum rebound test, cyclic parallel plate compression tests and high pressure compression set tests. All of the tests were performed on the O-rings directly. Hence, little preparation is required. The pendulum rebound tests were the most successful. They can be automatically interpreted to obtain estimates for the seal dynamic properties. The cyclic parallel plate compression tests are interpreted manually; otherwise, the tests are as useful as the pendulum rebound tests. The high pressure compression set tests were difficult to interpret and provided little insight into the properties influencing the O-ring life.

2.0 INTRODUCTION

The primary technical objective of this program was to develop a predictive method for seal life, using measurements of critical initial or aged physical or mechanical properties of seals, that allow an accurate estimation of actual service life of military specification hydraulic, lubricating oil and fuel seals. The primary emphasis is to develop predictive methods for elastomeric O-ring seals and other elastomeric seal designs operating within the -54 to 177°C (-65 to 350°F) temperature range and at pressures within 0 to 34.5 MPa (0 to 5000 psi).

Input parameters to the predictive model include seal gland and O-ring geometry, pressure, seal material properties, and loading. The predictive method can be calibrated by measuring the aged properties of seals recovered from operational service as well as artificially aged seals. The test method selected assesses the stress relaxation and/or other changes in dynamic properties (storage and loss moduli, for example) that have occurred as a result of compression of the seal within the seal gland. The test method was selected from methods such as force/deflection evaluation of the O-ring or a dynamic relaxation experiment to assess the changes of viscoelastic properties within the seal. *The selected test methods can be performed using standard quality control laboratory equipment with a minimum of required modifications.*

The predictive models require physical property measurements and the life models require calibration. Hence, the life model must be compared to experience and experimental results, receiving further refinements over time.

Current military specification elastomeric seals such as MIL-P-25732 and MIL-P-83461 nitrile (butadiene/acrylonitrile) hydraulic seals and MIL-R-25988 fluorosilicone fuel seals have high compression set (permanent deformation) on long term aging in fluid. The high compression set, along with high temperatures in operation, and dynamic wear, cause the seal to leak excessively and require replacement. The problem is aggravated for nitrile hydraulic seals by the inclusion of a plasticizer in the seal formulation to obtain the -54°C low temperature flexibility. As this plasticizer is eventually replaced by ambient fluid in the seal, low temperature leakage results. Fluorocarbon elastomers such as MIL-R-83248 military specification seal have better long term compression set but lack the low temperature flexibility to meet a -54°C requirement. Military specification requirements are based on short term, 70 hour compression set tests in hydraulic fluid or fuel. This single point test is inadequate to separate good seals supplied by seal manufacturers from marginal seals and there has been no correlation developed that will accurately predict the actual service life of these seals.

The reliability of military aircraft fluid systems is generally limited by the capability of elastomeric seals such as O-rings to maintain effective sealing force. The actual sealing force experienced by the seal is a complex function of the initial squeeze of the seal in the gland, swelling experienced by the seal as a function of temperature and pressure, and the compression set of the seal. In order to anticipate service life limitations, it is necessary to predict the time period over which these seals will continue to perform their function. The widely accepted parameter used to evaluate the condition of used seals is compression set. Curves, such as those contained in Figure 2.1 have been used to estimate the life of seals of various materials which are subject to varying thermal environments. An example of compression set measurements for O-rings is illustrated in Figure 2.2. However, work performed by United Technologies has shown a poor correlation of compression set measured with actual field service time accumulated by elastomeric seals. Results of eight MIL-R-25988 and MIL-R-83248 seals were taken from each of several components that had been removed from aircraft with varying numbers of flight hours. The data showed a very poor correlation between compression set and flight hours.

It is, therefore, necessary to develop a predictive method for seal life based on dynamic properties which will provide an improved correlation with service hours. This method can be developed with any military specification elastomer, such as MIL-R-25988. With the determination of additional material properties, the method could then be transferred to other military specification elastomers, such as MIL-P-25732 and MIL-P-83461.

The major variables which influence the life of an elastomeric seal are temperature, seal material, fluid, seal squeeze, pressure, and gland design. The model to be developed must predict the life of a seal based on these input variables. In order to correlate the model with actual service conditions, it is desirable to assess the condition of the seals at intermediate stages within the overall life.

Two types of variables affect seal performance, dimensional variables and basic material properties. Dimensional variables directly influence the ability of the seal to mechanically close the sealing gap and establish the required sealing force. Changes in dimensions, such as creep, result in distortion and the relaxation of the seal. These effects can and will compromise the ability of the seal to close the seal gap and maintain sealing force. Shifts in material properties are more subtle, but affect sealing performance nonetheless. Changes in the stiffness of the material and response to the applied forces will affect the performance of the seal. Dimensional changes are easy to measure, while shifts in properties are difficult to determine. Despite these difficulties, development of methods of measuring, analyzing and predicting properties shift would be of significant value. Seals obtained from different manufacturers and made to a common specification are not usually the identical seal formulation. As a result, the response to applied conditions may not be the same as well as properties changes as the seal ages. Development of test and predictive methods would allow selection of seals expected to give superior lifetime performance.

Many test methods exist for the determination of fundamental properties of elastomeric materials. These properties include the storage and loss moduli, which control the stiffness and damping of the elastomeric components. Poisson's ratio, the ability of the material to resist volume change as a load is applied, is another basic property. These properties are usually determined using specific test methods, such as Thermo Mechanical Analysis (TMA) or Rheological Dynamic Analysis (RDA). These instruments usually require sample coupons of a specific shape such as disks or flat bars. The analytical tools supplied with the instruments depend on analysis of specific shapes; this prevents analysis of finished products such as O-ring seals.

As mentioned previously, compression set has been used to assess the condition of elastomeric components with little success. The difficulty of using compression set and similar methods which measure external, geometric changes is that the method is insensitive to fundamental changes in material properties and behavior. Measuring the bulk distortion of test pieces provides no insight toward the strain distribution within the test piece and certainly no information about the distribution of stress in the material. Fundamental properties, such as storage and loss moduli, will change due to aging, exposure to solvent, and other environmental effects. The shift of these and other basic properties does affect the dynamic response of the parts and their ability to meet performance requirements. Test methods are required which can measure basic property shifts with pieces of differing geometries.

The solution to these limits is to develop shape-independent test methods and analytical tools to extract the material properties of components such as O-ring seals. Two methods were examined, developed and evaluated. One method attempted to measure and analyze multiple rebounds against a seal. A standard pendulum tester, typically called an Izod or Charpy impact test was adapted for this approach. In the second method, the O-ring was squeezed between two parallel plates, similar to the squeeze applied to an O-ring seal during typical operating conditions. This O-ring could be vibrated at a series of fixed frequencies and amplitudes while the force-displacement response was measured. Using analytical methods designed for the test, basic properties could be extracted from the data.

When used with new and aged seals, it was anticipated shifts in properties which might affect seal performance could be measured.

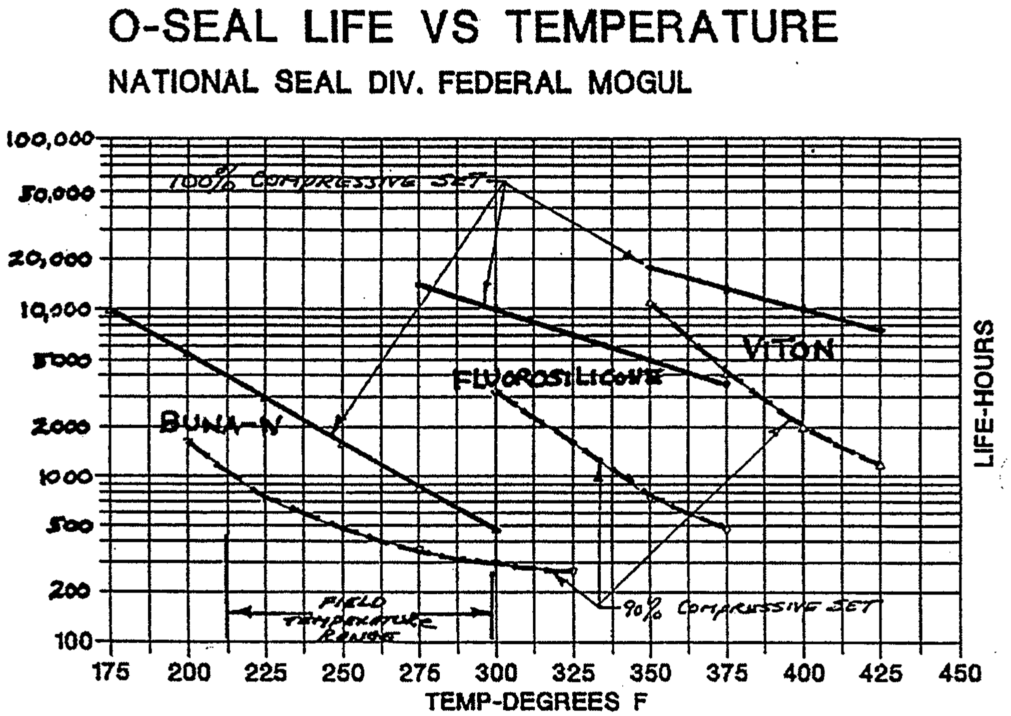
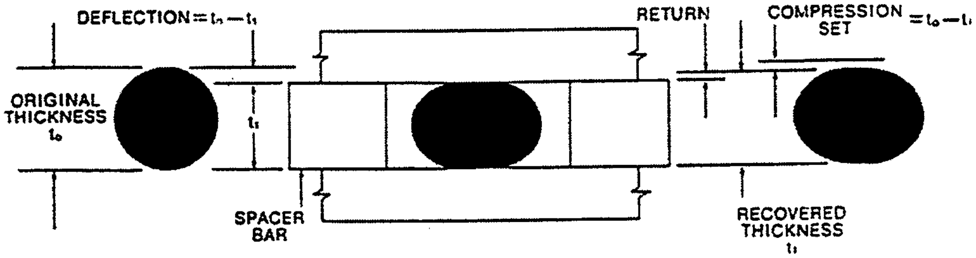


Figure 2.1 Temperature Effects on O-Ring Seal Compression Life



EXAMPLE: $t_o = 0.200$ $t_1 = 0.150$ $t_i = 0.190$

COMPRESSION SET (AS PERCENT OF ORIGINAL DEFLECTION)

$$C = \frac{t_o - t_i}{t_o - t_1} \times 100$$

$$C = \frac{0.200-0.190}{0.200-0.150} = \frac{0.010}{0.050} \times 100 = 20\% \text{ Compression Set}$$

(ASTM normally requires
deflection equal to 1/4 t_o)

Compression Set is generally determined in air and reported as the percent deflection by which an elastomer fails to recover after a fixed time under a specified squeeze and temperature. As expressed 0% indicates no relaxation has occurred whereas 100% indicates total relaxation; the seal just contacts mating surfaces against which it no longer exerts force. Compression set may also be stated as a percent of original thickness. However, percent of original deflection is more common.

Figure 2.2: Definition of Compression Set

3.0 APPROACH

During operation, properties of seals change as already mentioned. Over the service life, the seal must maintain adequate contact pressure to maintain a sealing as these properties change. Obviously, the primary failure mode of seals is leakage. Dynamic seals must control leakage under conditions where relative motion and wear occurs at the sealing interface. Under these conditions, the dynamic properties as well as static properties are critical to the performance of the seal. A method which could accurately assess changes of critical properties and predict the limits of satisfactory seal performance would minimize occurrences of in-flight failure by allowing effective and timely replacement of seals. Examples of critical properties and conditions include: dimensional changes caused by creep and compression set; physical property changes due to aging of the material; and material loss due to wear and abrasion.

Current military specification elastomeric seals such as MIL-P-25732 and MIL-P-83461 nitrile (butadiene/acrylonitrile) hydraulic seals and MIL-R-25988 fluorosilicone fuel seals have high compression set (permanent deformation) on long term aging in fluid. The high compression set, along with high temperatures in operation, and dynamic wear, cause the seal to leak excessively and require replacement. Military specification requirements are based on short term, 70 hour compression set tests in hydraulic fluid or fuel. This single point test is inadequate to separate good seals supplied by seal manufacturers from marginal seals and there has been no correlation developed that will accurately predict the actual service life of these seals.

Therefore it is necessary to develop a predictive method for seal life based on a dynamic property, such as stress relaxation, which should be more sensitive to material properties changes during aging and which will provide an improved correlation with service hours. This method is designed to be used with any military specification elastomer, such as MIL-R-25988, MIL-P-25732 and MIL-P-83461. For the development of the predictive method MIL-R-25988 and MIL-P-83461 seals were each selected from three different manufacturers. These seals were subjected to laboratory aging conditions which involved exposure to elevated temperature and fluid at a nominal operating pressure. The MIL-R-25988 seals were exposed to JP-8 jet fuel at 900 psi pressure and 121°C or 149°C. Similarly, the MIL-P-83461 seals were exposed to hydraulic fluid at 27.6 MPa pressure and 107°C or 135°C. At each temperature fluid aging was done for nitrile and fluorosilane seals for 168 and 504 hours. Following exposure, changes in the dimensions and weight of the seals were measured and the seals were sent for dynamic testing.

Development of a successful method for predicting seal life requires the following key elements:

- Predictive model of seal performance
- Selection of testing methods and measurement of critical properties of initial and aged samples
- Correlation and refinement of predictive model results with test data

Current life prediction methods are highly dependent on the compression set tests. These tests are difficult to interpret accurately, and are not dependable for predicting the life of a seal. A direct correlation between properties and aging may provide the designer with a more accurate prediction of the seal life. Based on these considerations two models were developed for predicting seal life: (1) a simple PC model for design calculations and (2) a more complex workstation model for detailed analysis. The PC model is discussed in Section 6.2 and the workstation model in Section 6.3.

The computational models require material test data to establish the parameters used in the mechanical response of the seal. Single point measurements, such as compression set and durometer (hardness), are insensitive to changes in material properties. Dynamic methods, such as measurement of storage and loss moduli by torsional rheology, are very sensitive to material properties shifts. However, while this method works well for rectangular test coupons, it is impractical for the testing of O-rings.

Tensile and compressive testing of O-rings to determine force-deflection response at different strain rates and temperatures should be sufficient. The effect of environmental aging of the materials can be established using initial and aged test coupons in a parallel testing effort. The dynamic properties of these samples, including storage (G') and loss (G") modulus can be determined by torsional rheology using a Rheometrics rheological dynamic analyzer (RDA).

4.0 MATERIALS SELECTION

Military specification seals were selected based on the service experience of the Air Force. Seals causing the most significant field problems were identified through consultation with the Materials & Manufacturing Directorate, Air Force Research Laboratory. The military (and commercial) aircraft experience of United Technologies Hamilton-Standard was also considered. The military specification seal types selected were evaluated using the life prediction method selected during Task 1.

MIL-P-25988 fluorosilicone fuel seals used in military and commercial fuel controllers were selected as one candidate for this program. Hamilton-Standard experience has shown these seals to be the current service life limiting factor in aircraft fuel system sealing applications. Leakage of aircraft fuel systems is particularly troublesome due to the potentially catastrophic consequences. MIL-P-25732 and MIL-P-83461 were considered as examples of nitrile hydraulic seals. Based on the experience of Hamilton-Standard, MIL-P-83461 seals were selected for evaluation.

To determine the effect of differences between seal compounds (materials) used by different manufacturers, each type of seal was obtained from three sources. Two sizes of MIL-R-25988 fluorosilicone seals, 25988/2-007 and 25988/2-214 were selected for testing and evaluation. A single size of the nitrile seal, MIL-R-83461/1-214, was selected for evaluation. In addition, MS28774-214, fluoropolymer backing rings were selected for use with the M83461 seals. The seal distributor, Sealing Solutions, obtained single batches of each type of seal from the manufacturers listed below. Certificate of conformance documents were obtained for each lot of seals to verify basic requirements.

MIL-R-25988 (fluorosilicone)	Parco Company Parker International Seal
MIL-P-83461 (nitrile)	Parco Company Parker Wynn's Precision
MIL-28774 (fluoropolymer)	Tetrafluor

JP-8 jet fuel was selected as the test fluid for the MIL-R-25988, fluorosilicone O-rings. The JP-8 fuel was obtained from the Air Force via Wright-Patterson labs. Brayco Micronic 882 (MIL-H-83282) hydraulic oil was selected as the test fluid for the MIL-R-83461 nitrile O-rings. Brayco Micronic 882 was supplied by the Hamilton-Standard division of UTC.

5.0 AGING METHODS

To develop methods for characterizing changes of properties in elastomeric materials such as O-rings, methods to artificially age samples were required under controlled, accelerated conditions. To develop the required aging capabilities, fixtures, environmental conditions and processes first needed to be defined for each O-ring material.

5.1 Test Conditions

Conditions were identified that simulated the actual operating environment that the O-rings would experience. For example, actual gland dimensions were selected to provide the proper squeeze on the seal. Exposure temperature, pressure and fluids were selected to simulate real world conditions. A second set of conditions with increased test temperature was selected to accelerate the process. The following conditions were selected:

Seal	Size	Temp (°C)	Pressure (MPa)	Fluid	Time (hrs)
MIL-P-25988	-007, -214	121	6.2	JP-8	168
MIL-P-25988	-007, -214	121	6.2	JP-8	504
MIL-P-25988	-007, -214	149	6.2	JP-8	168
MIL-P-25988	-007, -214	149	6.2	JP-8	504
MIL-P-83461	-214	107	27.6	Brayco Micronic 882	168
MIL-P-83461	-214	107	27.6	Brayco Micronic 882	504
MIL-P-83461	-214	135	27.6	Brayco Micronic 882	168
MIL-P-83461	-214	135	27.6	Brayco Micronic 882	504

JP-8 fuel and Brayco Micronic 882 hydraulic oil, as stated previously, were the test fluids.

5.2 Test Fixtures

Test fixtures were required that were capable of subjecting test O-rings to the conditions described above for the required test period. Fixtures were designed using the gland dimensions for the –007 and –214 O-rings described by the SAE Aerospace Standard, AS4716. The fixtures were designed to each hold 8 O-rings. Examples of the test fixture designs are illustrated in Figures 5.1 and 5.2.

Each fixture was assembled in a pressure vessel to provide containment and these were attached to a manifold which was constructed in a Blue M, Model IGF 6680F-4, forced air oven. The manifold was constructed using stainless steel 304 tubing and pressure fittings capable of handling 34.5 MPa at temperatures above 149°C. Pressure was supplied by an Enerpac P-39 hand pump capable of supplying a maximum of 68.9 MPa of pressure. The assembly is illustrated in Figures 5.3 and 5.4.

5.3 Aging Procedure

The aging procedure of the test seals required well-defined methods for installation, measurement and aging of the test seals to ensure repeatability of the procedure. These methods are described below.

Prior to installation on the test fixtures, O-rings were examined for defects and deformities. Those showing obvious damage were rejected. The thickness and width of each O-ring was measured at 3 points using a digital micrometer and the average taken. The weight of each O-ring was measured to 4 decimal points. Each O-ring was given a unique ID relating to supplier, aging conditions and position in the aging fixture and the data was recorded.

O-rings were assembled wet with test fluid onto the aging fixtures, shown in Figure 5.1 and 5.2, using mounting tools designed to minimize damage to the O-rings. The mounting tools are shown in Figure 5.5. If an O-ring was damaged during assembly, a substitute was used and the above data revised to reflect the change. During assembly of the M-25988/2-007 O-rings, breakage of about 15% of the test pieces was noted. After the assembly of the O-rings, the collars were assembled onto the fixtures and the assemblies were placed into the containment vessels. The completed test fixtures were then connected into the pressure manifold inside the Blue M forced air furnace.

The test system was evacuated using a mechanical pump to remove air from the system for 5 minutes and was then backfilled with test fluid. (JP-8 or Brayco 882) The system was pressurized to full test pressure at room temperature to check for leaks. When the integrity of the system was confirmed, the pressure was lowered to 100 psi and the oven was programmed to heat to the test temperature. During the heating cycle, typically 8 hours, the test pressure was checked periodically to avoid an over-pressurization condition. When the test temperature was reached, the test pressure was checked every 4-12 hours and adjusted to maintain the test pressure within 10%. Tests were continued for the programmed test time, followed by an overnight cooling of the oven with the samples under pressure.

After cool down, residual fluid pressure was relieved and the test fixtures were removed from the oven. One fixture was disassembled at a time and the O-rings were removed, dried, measured and weighed. Sample recovery and measurement occurred in the 6 hours following depressurization. Following measurement each O-ring was stored in a marked, fluid filled polyethylene bag prior to testing. Samples were usually taken for rebound testing within 48 hours of collection and dynamic modulus testing within 7 days of collection.

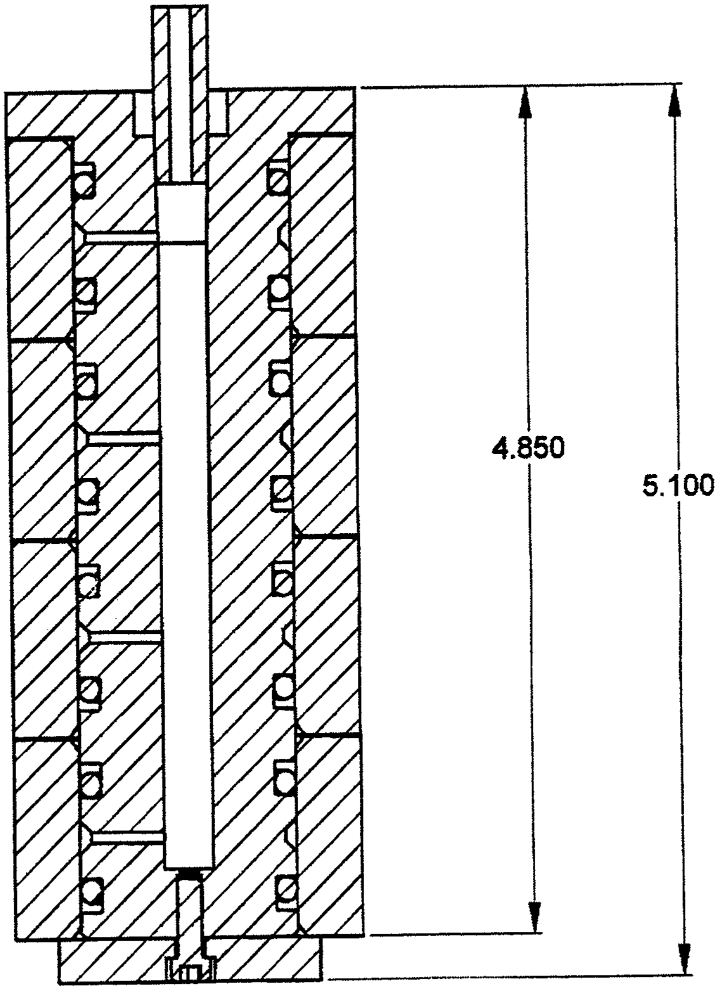


Figure 5.1: Test Fixture for Aging of O-Ring Seals

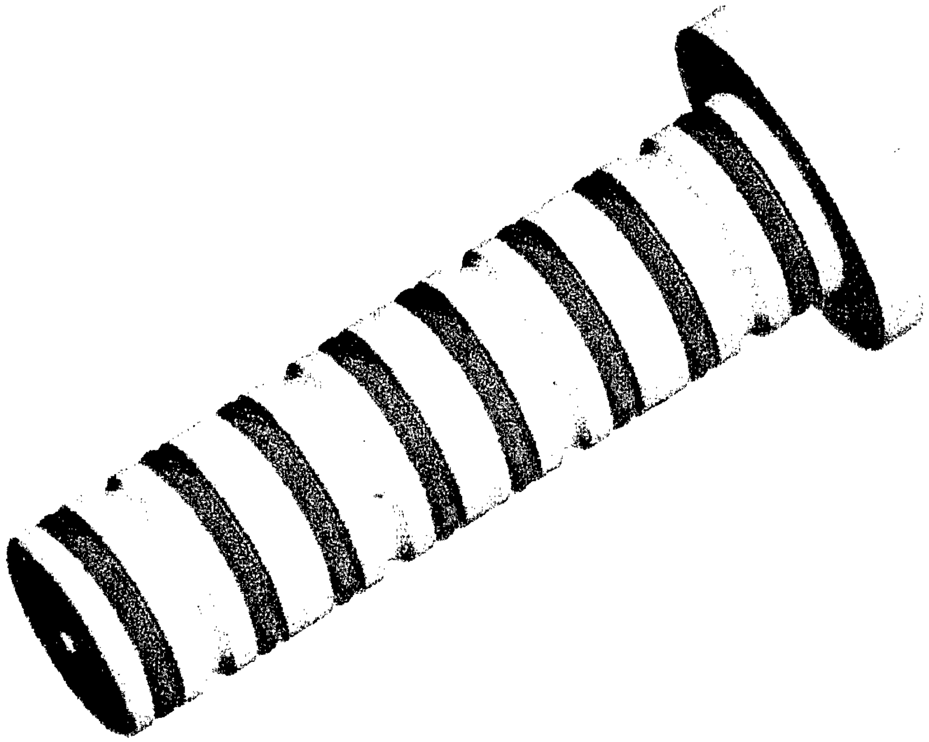


Figure 5.2: Test Fixture for Aging of O-Ring Seals

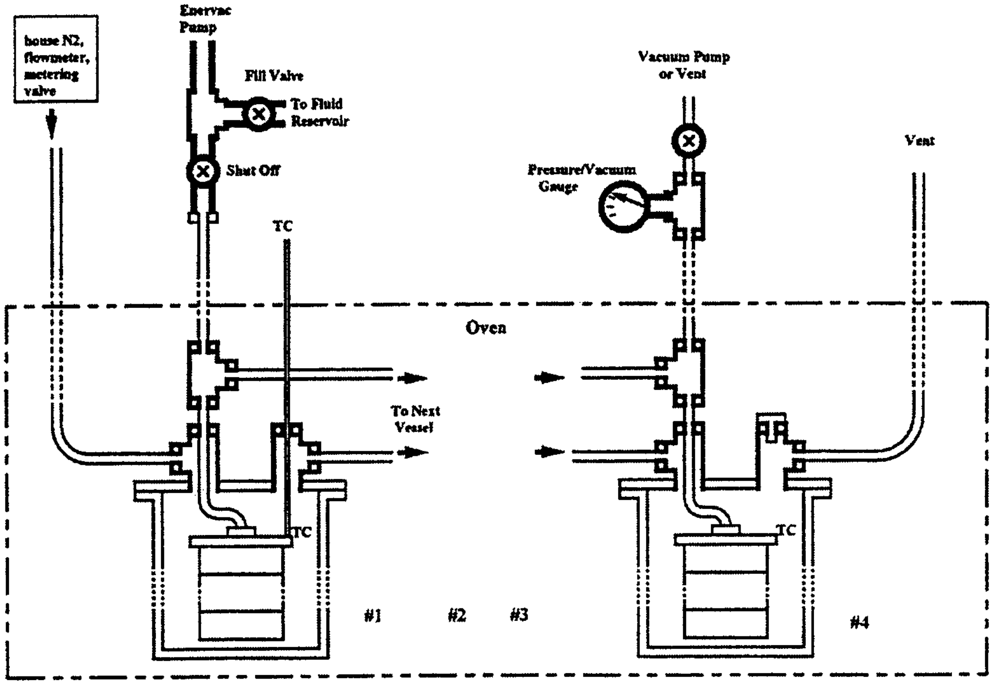


Figure 5.3: Aging Assembly for O-Rings

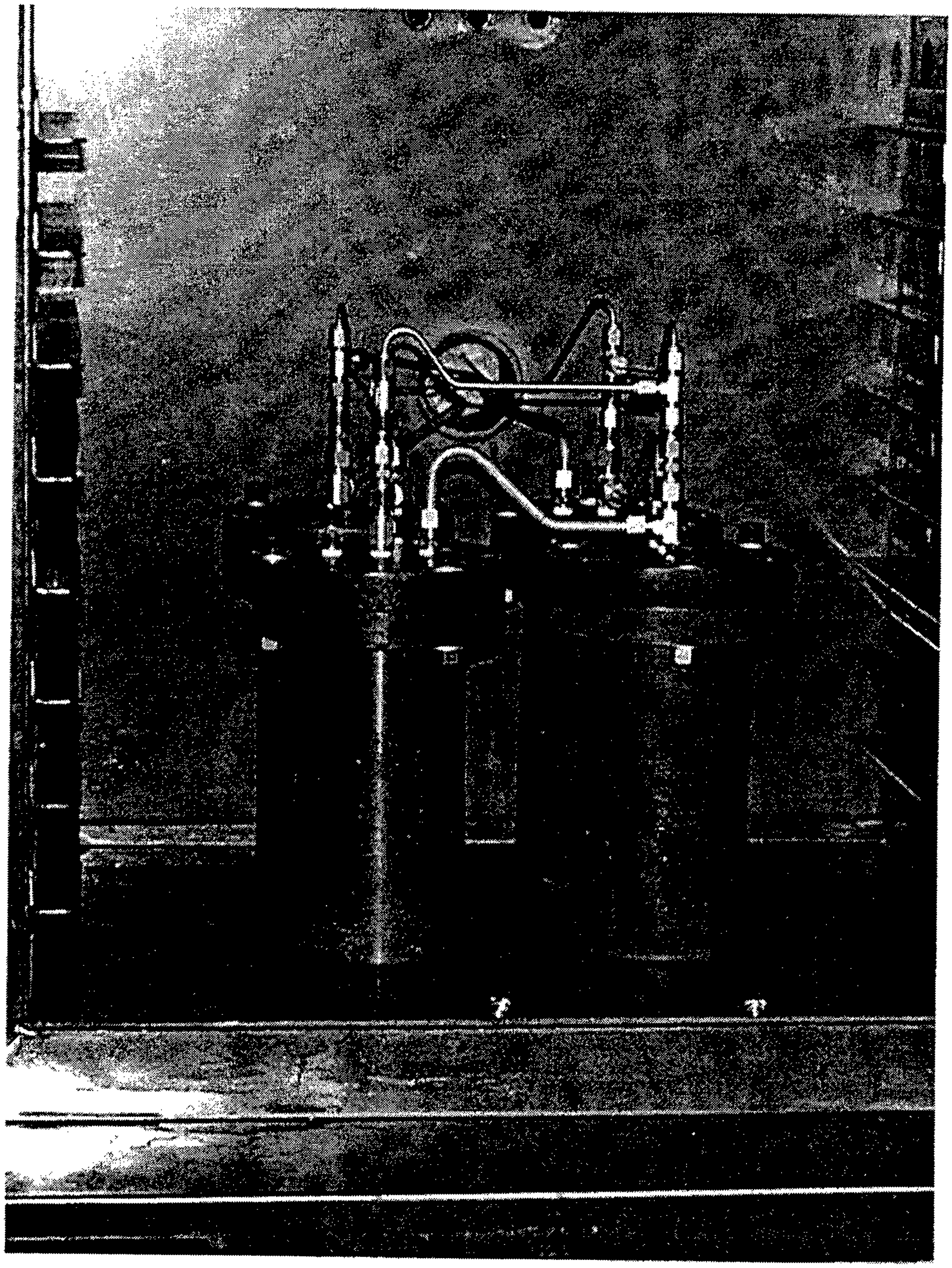


Figure 5.4: Aging Assembly for O-Rings

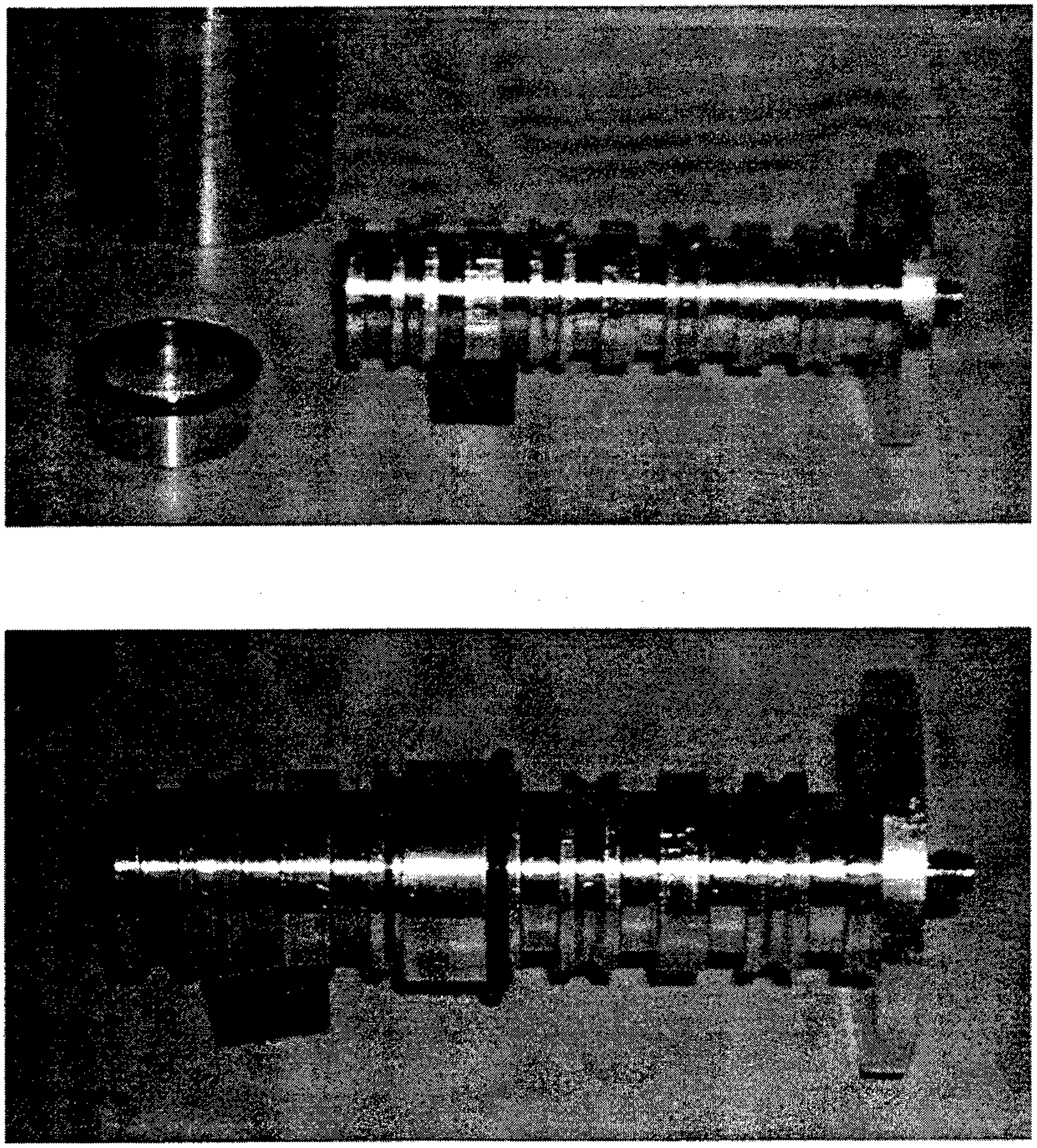


Figure 5.5 Assembly Tool for Mounting O-Rings in Aging Fixture

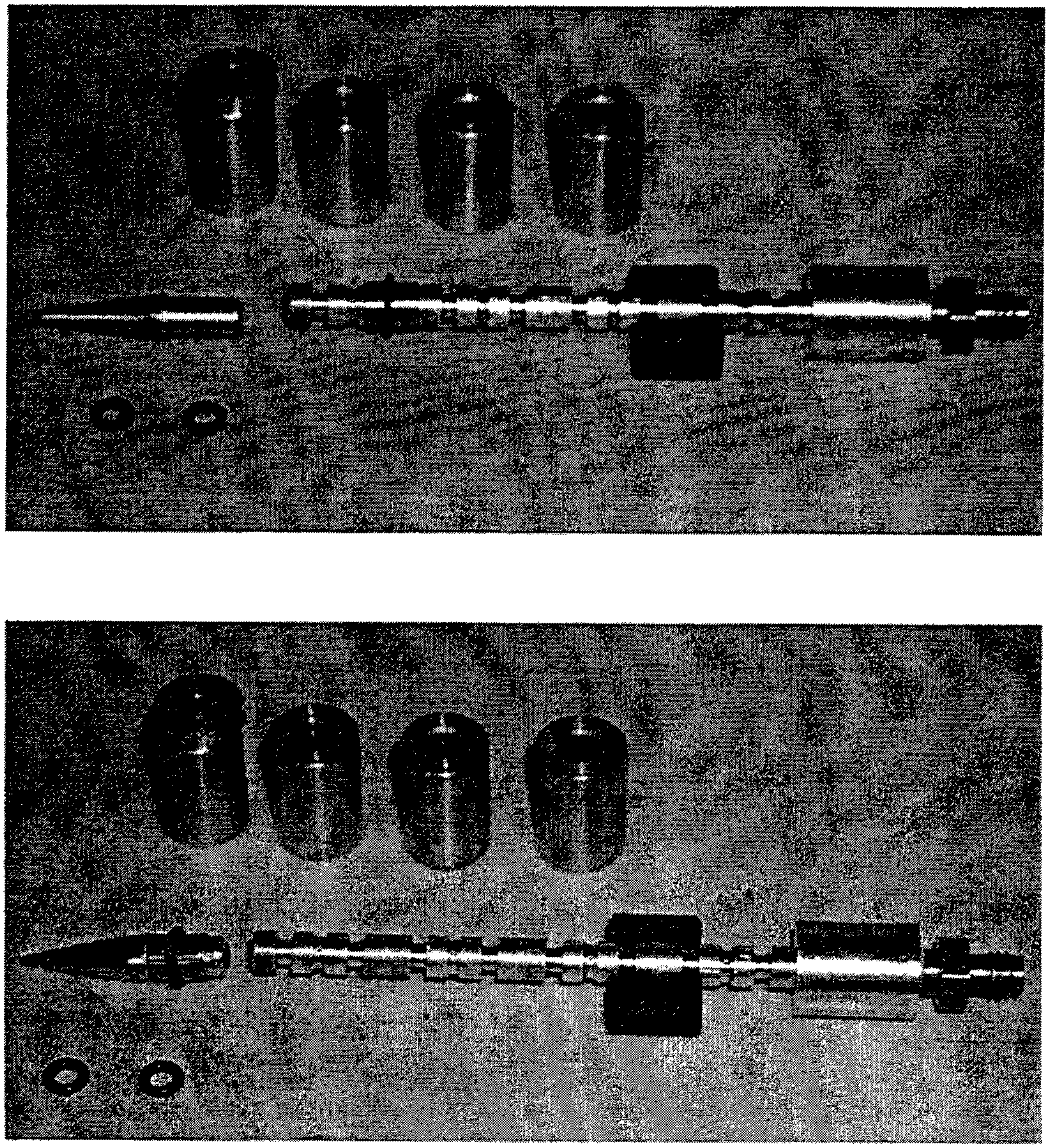


Figure 5.6 Assembly Tool for Mounting O-Rings in Aging Fixture

6.0 LIFE PREDICTION METHOD

Many levels of modeling are available in the analysis of polymeric seals. The material can be modeled as elastic using large strain rubber elasticity. Linear damping effects can be included by using a viscoelastic model. Nonlinear, time dependent effects can be included by using creep models, and all of the above can be combined using viscoplastic models.

Large strain rubber elasticity has been understood for over fifty years[1,2]. Polymers are generally assumed to be isotropic. The response is usually described in the principal strain directions and the strain measure is taken to be the stretch ratios. Each of the principal stresses are then found by taking the derivative of the elastic potential with respect to the strain components, where the elastic potential is given by

$$W = \frac{G}{2}(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - 3) \qquad (1)$$

The λ_i are the principal stretch ratios, $i=1,2,3$, and G is the shear modulus. In rubber elasticity the volume is conserved, or

$$\lambda_1\lambda_2\lambda_3 = 1 \qquad (2)$$

which makes the Poisson ratio, ν, one half (for small strains and under ideal conditions). For a Poisson ratio of one half the shear modulus and Young's modulus are related by

$$E = 3G \qquad (3)$$

Additional terms have been added to the elastic potential and are commonly incorporated. In particular the Mooney-Rivilin[1] model is common,

$$W=C_1(I_1 - 3) + C_2(I_2 - 3) \qquad (4)$$

where

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \qquad (5)$$

$$I_2 = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 \qquad (6)$$

are invariants. For many polymers an adequate approximation of the mechanical response results from

$$C_1 = G/2 = E/6 \qquad (7)$$

$$C_2 = C_1/4 \qquad (8)$$

The formulations above require that only the Young's modulus for small strains must be measured. The fact that the material is incompressible, (i.e. the volume is conserved) means that the hydrostatic component of the stress tensor cannot be evaluated directly from the strains, and hence becomes an additional unknown in the governing equations. This additional unknown is balanced by the assumption that the material is incompressible. A better match between measured response can be achieved by the addition of higher powers in the terms I_1-3 and I_2 -3.

Viscoelasticity is commonly used to approximate the time dependent response of polymers. Normally the constitutive relation is written in integral form[3],

$$\varepsilon_{ij} = \int_{-\infty}^{t} J_{ijkl}(x,t-\tau)\frac{\partial\sigma_{kl}}{\partial\tau}d\tau \qquad (9)$$

or equivalently

$$\sigma_{ij} = \int_{-\infty}^{t} G_{ijkl}(x,t-\tau)\frac{\partial\varepsilon_{kl}}{\partial\tau}d\tau \qquad (10)$$

where G_{ijkl} are the moduli, J_{ijkl} are the compliances, x is the position vector, t is the time, $\square_{ij}$ is the strain tensor, and σ_{ij} is the stress tensor. The moduli are generally taken in the form of an N term Prony series[4],

$$G_{ijkl} = G_{ijkl}^{\infty} + \sum_{n=1}^{N} G_{ijkl}^{n} e^{-t/t^{n}} \qquad (11)$$

where G^{∞}_{ijkl}, G^{n}_{ijkl} and t^{n} are material parameters.

Permanent set in polymers is not generally a linear effect, and it may not be possible to represent it using viscoelastic equations. Such representations may require the use of a nonlinear creep theory. Polymer creep models are generally borrowed from those used for metals. Generally a power law is assumed,

$$\dot{\varepsilon} = A\sigma^{n} \qquad (12)$$

and the creep deformation is assumed to be incompressible. Hence, the constants A and n in equation (6) are sufficient for characterizing the three dimensional response of the isotropic material, and are generally temperature dependent.

The above formulations separate the nonlinear time dependent (creep) and nonlinear time independent (plasticity) response of the material, but at sufficiently high temperatures plasticity and creep become difficult to separate. Viscoplastic constitutive laws have been developed, based on the concept of overstress. The concept of overstress was first defined by Krempl[6], and recently extended to large strains[7]. These extensions have been shown experimentally to be applicable to polymers.

6.1 Material Response Description

The viscoelastic properties for polymers are usually represented by a storage (shear) modulus, *G'*, and a loss (shear) modulus, *G''*. These are found by measuring the torsional load required for an enforced sinusoidal twist of a thin rectangular cross-section. The ratio of the in-phase torsional shear stress and the corresponding shear strain gives the storage modulus, while the ratio for the out-of-phase stress and strain gives the loss modulus. For an isotropic material represented by equation (10) only one stress and one strain component are present. Under these conditions equations (10) and (11) become, respectively,

$$\tau = \int_{-\infty}^{t} G\ (x,t-t')\dot{\gamma}(t')dt' \qquad (13)$$

and

$$G = G^{\infty} + \sum_{n=1}^{N} G^{n} e^{-t/t^{n}} \qquad (14)$$

where $\dot{\gamma} = d\gamma / dt'$.

Assume an imposed sinusoidal stain

$$\gamma = \gamma_0 \sin \omega t \tag{15}$$

where γ_0 is the shear strain amplitude, and ω is the frequency. Substituting equations (14) and (15) into equation (13) results in

$$\tau(t) = G^{\infty}\gamma_0 \sin \omega t + \sum_{n=1}^{N} G^{n}\gamma_0 \{\frac{\sin \omega t}{1+(\omega / t^{n})^2} + \frac{1/(\omega / t^{n})\cos(\omega t)}{1+(\omega / t^{n})^2}\} \tag{16}$$

The storage and loss moduli are given by the coefficients of $\gamma_0 \sin \omega t$, and $\gamma_0 \cos \omega t$, respectively.

Then

$$G' = G^{\infty} + \sum_{n=1}^{N} \frac{G^{n}}{1+1/(\omega t^{n})^2} \tag{17}$$

and

$$G' = \sum_{n=1}^{N} \frac{G^{n}/(\omega t^{n})}{1+1/(\omega t^{n})^2} \tag{18}$$

Note that for very low frequencies

$$G' = G^{\infty} \qquad \text{and} \qquad G'' = 0 \tag{19}$$

while for very high frequencies

$$G' = G^{0} \qquad \text{and} \qquad G'' = 0 \tag{20}$$

where $G^0 = G^{\infty} + \sum_{n=1}^{N} G^{n}$.

Hence, G'', must have at relative maximums at some frequencies. This can be found by setting the derivative of G'' with respect to frequency equal to zero. The result is G'' is a relative maximum when

$$\sum_{n=1}^{N} \frac{G^{n}[1-(\omega t^{n})^2]}{[1-(\omega t^{n})^2]^2} = 0 \tag{21}$$

For one term in the Prony series, the frequency at which the maximum in G'' occurs is at

$$\omega = 2\pi f = 2\pi / t^{1} \tag{22}$$

Relaxation tests can be used to find the parameters in equation (14). In a relaxation test the torsional strain is suddenly applied, and then held. The torsional stress, required to hold the strain, is measured as a function of time. The initial measure of stress to strain is an estimate for G^0 in equation (20). The long term, steady state ratio of stress to strain is G^{∞}. The parameters G^n

and t^n in the Prony series can be found by: least squares techniques, the use of inverse Laplace transforms, or by trial and error.

Long term creep tests can be analyzed in a similar manner. For example, consider a Prony series consisting only of the term n equal to one. If a shear stress , τ_0, is suddenly applied at time equal to zero and then suddenly released at time t_0, the strain history is given by

$$\gamma = \begin{cases} \frac{\tau_0}{G^1}(1+\frac{t}{t_0}) & 0 < t < t_0 \\ \frac{\tau_0}{G^1}(\frac{t}{t_0}) & t_0 < t < \infty \end{cases} \tag{23}$$

Both relaxation and creep data reduction can be useful for the initial interpretation of various compression set tests. Since compression set tests are sometimes performed at higher temperatures to accelerate the tests, we must be able to project the results to lower temperatures. This can be accomplished by noting, if the mechanism producing the permanent strains has not changed over the temperature range of interest, then the rates will be proportional to *exp(-Q/RT)* (where Q is a heat of reaction, R is the gas constant, and T is the absolute temperature.)

6.2 PC O-Ring Code

Salita has developed a simple finite element based code for NASA[8,9] to aid in describing the response of the Shuttle solid rocket O-ring seals. The code geometric input can be used to describe in general piston and rod seals, as shown in Figure 6.1. Face seals (Figure 6.1), though, can only be approximately modeled. The code includes the ability to model transient gap motion and pressure transients. The material model, although approximate, can include time dependent material response.

However, the computer code was designed to simulate the Space Shuttle solid rocket seals during launch; thus it included specific pressure and gap launch transients. Those portions of the code had to be modified to include more general transient loads. Three loading functions were added. Each was classified according to a type. Type 0 ran the program with a suddenly applied pressure after the gap was closed. This was a test mode the program originally contained. Type 0 also suddenly changed the gap simultaneously. In type 1, the gap and pressure change after a specified time, and then revert back to the original state an additional time later. The most useful loading type is type 2, which includes a sinusoidal variation in the pressure and the gap. The pressure variations with respect to the gap can only be in phase or 180 degrees out of phase. The last type, type 3, includes an exponential decay or rise in the pressure and gap. Again, for type 3, the pressure and gap change by the same ratio at all times. Several other minor changes were made to accommodate a more general O-ring design analyses.

The modified code can be used on a workstation or on a PC. On the PC there are two versions: a Visual Basic version and a DOS version. The DOS version will run on any PC that has MS-DOS or a DOS prompt, but requires the user to edit a file containing the input to run the program. A sample input file is shown in Figure 6.2. The Visual Basic version, which should run on most PC's has labeled entries for each of the parameters shown in Figure 6.2. The first entry in the data is a title for the case being run. The geometric inputs for the PC version include: gap, gland width, gland height, offsets at the sides of the gland, the cross-section diameter of the O-ring, the inside diameter of the O-ring gland, the unstretched diameter of the O-ring. The material properties include the Young's modulus, the relaxation time and the relaxation factor. Loading includes the applied pressure, the back pressure, and the coefficient of friction between the gland and the O-ring. The loading also includes a type (discussed above) together with the variations in the applied pressure and the gap, and the time period for the loading. The last entries include the number of time steps and the time step.

The output resulting from Figure 6.2 is illustrated in Figures 6.3 through 6.5. Figure 6.3 displays the deflected shape at a particular transient time. Figure 6.4 plots the transient history of the footprint of the O-ring on the walls of the gland, while Figure 6.5 gives the transient normal force history on each wall. If friction is present, then a plot of the tangential wall forces can also be obtained.

There are several limitations to the PC code. The code was developed on workstations and main frames using Fortran. When moved to the PC a Fortran to C translator was used and the C code was compiled. Hence, when an error occurs during execution (usually this occurs because of input data problems) the message only references locations in the C code, if the debugger is used. This means that code errors can only be accurately traced on a computer where the Fortran is compiled directly. A Fortran compilation on a PC would remove this problem. If an error occurs with a C compiled version then the most likely error is an O-ring configuration that cannot react the loads against the walls. This will happen when a large gap develops between higher and lower pressure regions in the gland. The PC version uses the graphic interface package Grafic. The PC version of Grafic is still experimental and cannot make hardcopies of the deformed shape, but does store postscript files of the load and footprint histories. Finally, if friction is present, it may take an extra time step to release nodes from a wall.

A complete description of the Visual Basic version is included in Appendix A, and similar discussion for the DOS version is in Appendix B. Each Appendix is self-contained so that they can be copied and separated from the report.

6.3 Workstation Codes

Two versions of finite element based workstations were developed. One to be used with MARC and the second with ANSYS. Each uses a minimum amount of input from the user.

6.3.1 MARC Version

The MARC version is based on the nonlinear finite element code distributed by MARC Analysis Research Corporation in Palo Alto, California. It includes contact and friction formulations along with large strain rubber viscoelasticity as in equations (4) and (13). A general Prony series can be included as in equation (11). The transient motion of the contact surfaces (i.e., the gland surfaces) and the pressure history can be quite general. The code developed for use with the

MARC version includes a Fortran routine that generates the MARC input from user inputs. The user is either prompted for the inputs, or the program can be run using a redirected input file.

Figure 6.6 is an example of a redirected input file. The inputs include a title followed by the inside and minor O-ring diameters. The material properties include the storage modulus, *tanδ* (i.e., the ratio of the loss modulus to the storage modulus) and the frequency where the loss modulus is a maximum. The friction coefficient between the O-ring and the gland is entered next. The next three sets of entries describe the O-ring geometry. The next several entries describe the loading. The applied internal pressure and the external (back) pressure are entered next, followed by the mean and maximum compression squeeze. The squeeze is defined as the change in cross-section diameter divided by the cross-section. The next entries are the pressure change over one cycle and the phase angle with respect to gland surface motion. The last two sets of entries include the number of cycles to simulate, the frequency of oscillation, and the number of load steps for the gland motion, pressurization, and the number of load steps per cycle.

MARC user subroutines were written for the gland motion and pressure loading history. A mesh was generated and stored. This mesh is then used for the runs that are generated. A Unix script has been written that automatically runs the program to make the MARC finite element model and then run the MARC finite element code.

Figures 6.7 through 6.13 illustrate some of the results that can be obtained. Figure 6.7 is a face seal that is subject to transient motion of the gland faces. Figures 6.8 and 6.9 illustrate the stresses and the deflections at two different times with nodes and elements shown. Figure 6.10 shows a rod seal while Figure 6.11 demonstrates the deflections and stresses at a later time with only elements shown. Finally Figure 6.12 illustrates a piston seal. Figure 6.13 displays the stresses and deflections at a later time without any mesh definition included. Note that this last case does not display the current location of the gland surfaces.

The MARC version readily runs cases with gland motion but can have numerical instability problems with large pressure loading. This instability can represent a real physical instability or could be numerical in nature. A buckling analysis can aid in determining the source of the instability.

A complete description of the MARC version is included in Appendix C. The Appendix is self-contained so that it can be copied and separated from the report.

6.3.2 ANSYS Version

The ANSYS version cannot simulate the diverse problems that the MARC version can, but the ANSYS version can readily model the parallel plate cyclic tests. To perform these simulations, an ANSYS 5.4 macro was written which prompts the user for geometric and material data, generates a model of a viscoelastic rubber O-ring, and runs a compression and decompression cycle using rigid compression surfaces. This simulation is particularly useful for interpreting the parallel plate tests.

After entering ANSYS, the user needs to type oring at the command line which executes the macro oring.mac. The user is then prompted for a job name and title. Geometric data is then requested (see Figure 6.14) which includes the major diameter and minor diameter. The major diameter is the inner diameter plus half the cross-section diameter. Loading requirements are entered next. These include the compression displacement and compression velocity. Material properties requested include the instantaneous shear modulus, G^0, the infinite time shear modulus, G^∞, and the relaxation time, t^1.

Once the user completes the entry of geometry and material variables, the macro generates the geometry and the mesh, defines the rigid surfaces, and executes three load steps. Several material parameters are currently set by the macro (although they can readily be modified by editing the macro). These parameters are as follows:

G = G(inf.)	(The global elastic shear modulus is set equal to G(inf.) in the viscoelastic model)
$\nu = 0.475$	(Poisson's ratio)
$\mu = 0.0$	(Friction coefficient - Zero friction assumed)

The load steps are set up to run a compression, hold, and decompression:

ANSYS returns a warning during the simulation indicating that the viscoelastic elements have not been tested with the large strain option. The test cases completed indicate this should not be a problem.

A complete description of the ANSYS version is included in Appendix C. The Appendix is self-contained so that it can be copied and separated from the report.

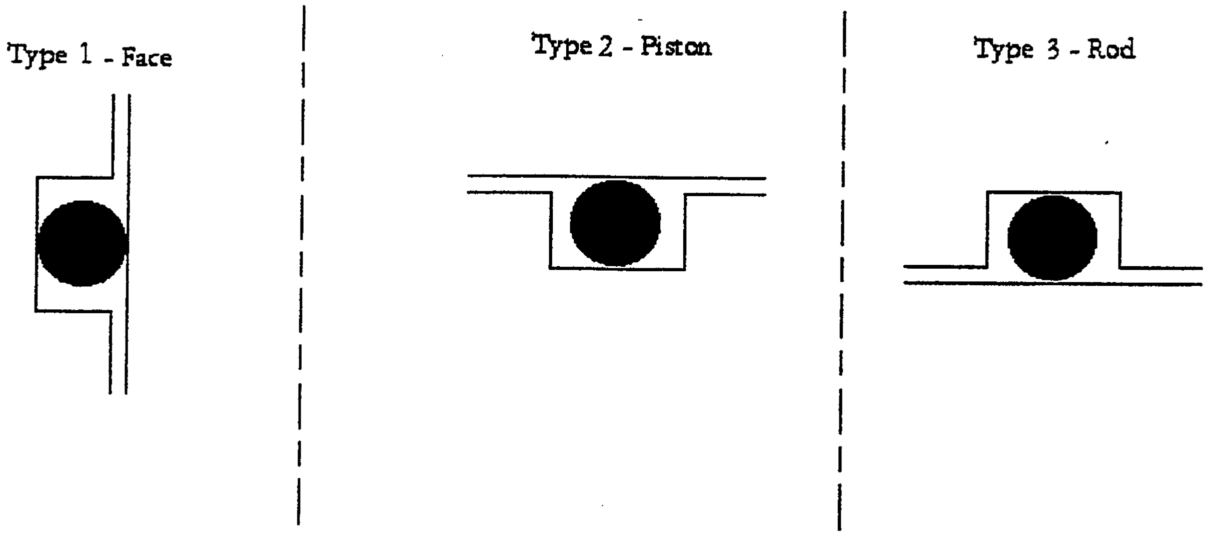
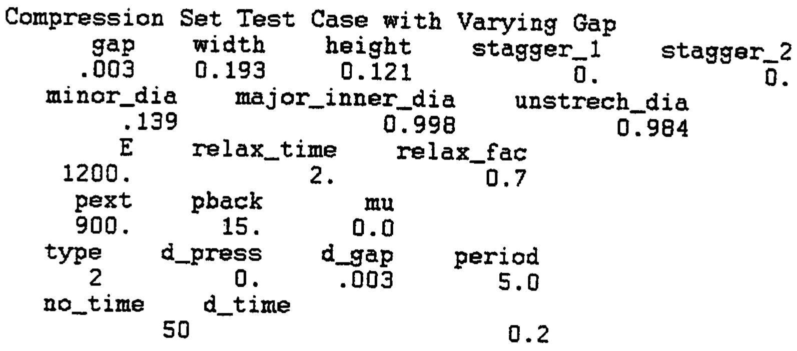


Figure 6.1 Typical O-ring Configurations



```
Compression Set Test Case with Varying Gap
      gap    width     height    stagger_1    stagger_2
     .003    0.193      0.121           0.           0.
   minor_dia     major_inner_dia     unstrech_dia
        .139               0.998             0.984
        E     relax_time     relax_fac
    1200.           2.            0.7
     pext     pback        mu
     900.       15.       0.0
   type     d_press     d_gap     period
     2          0.      .003        5.0
   no_time     d_time
          50                        0.2
```

Figure 6.2 Example PC-DOS Life Prediction Input

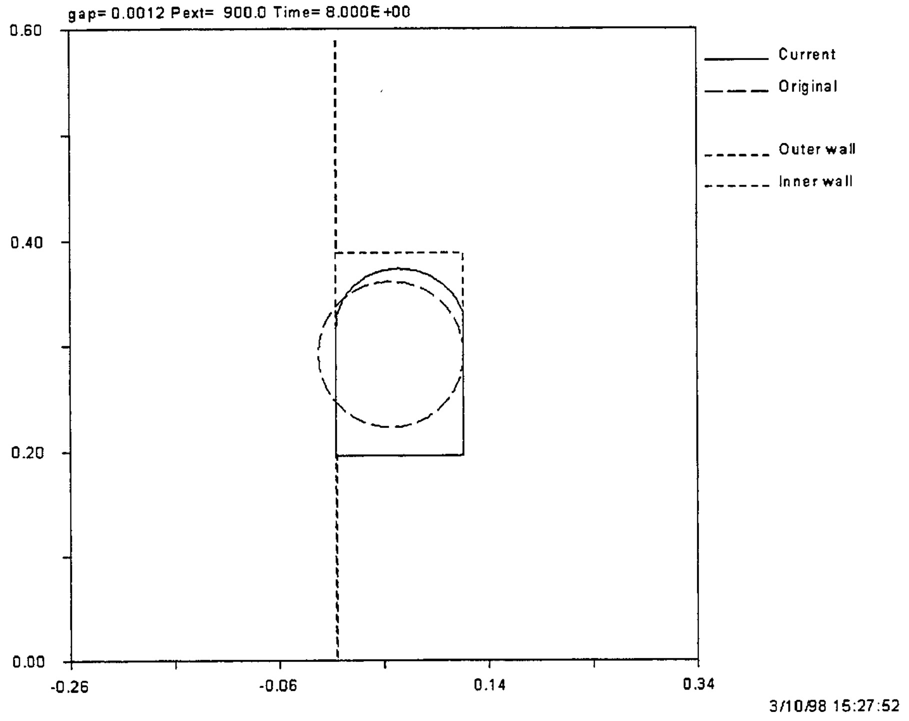


Figure 6.3 Compression Set Test Case with Varying Gap
gap = 0.0012 Pext = 900.0 Time = 8.000E+00

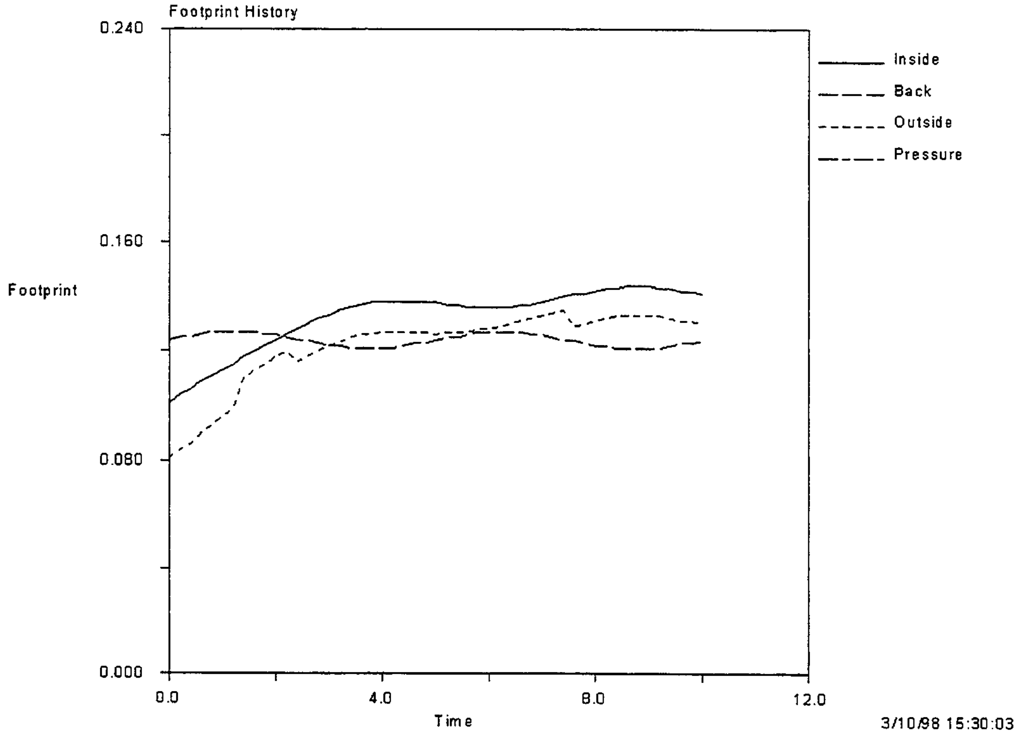
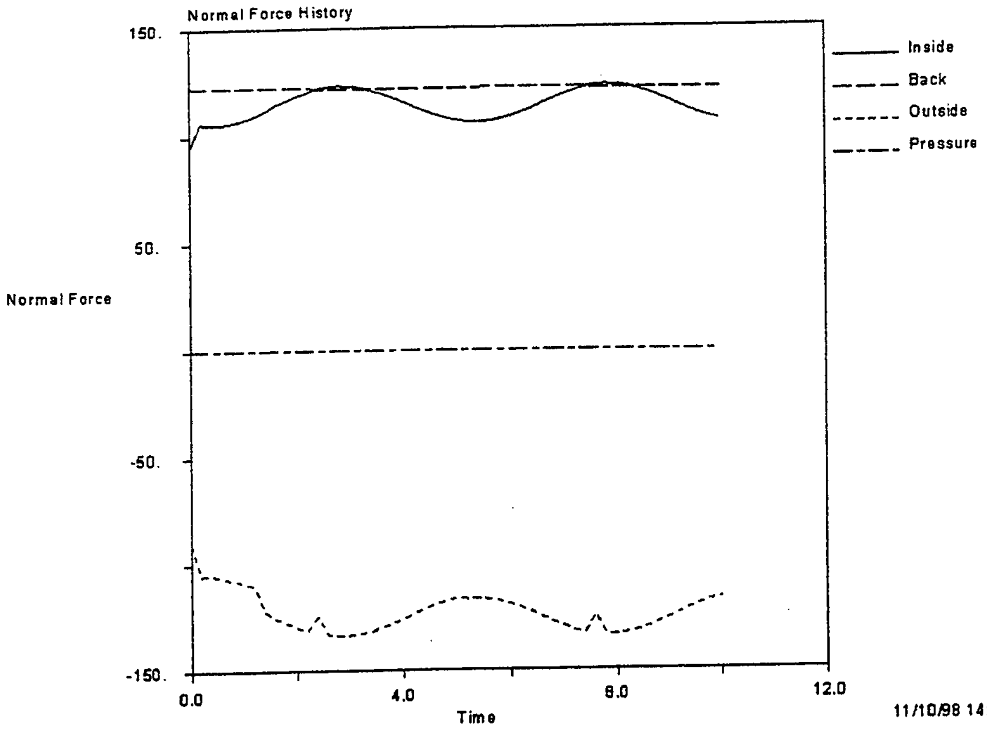
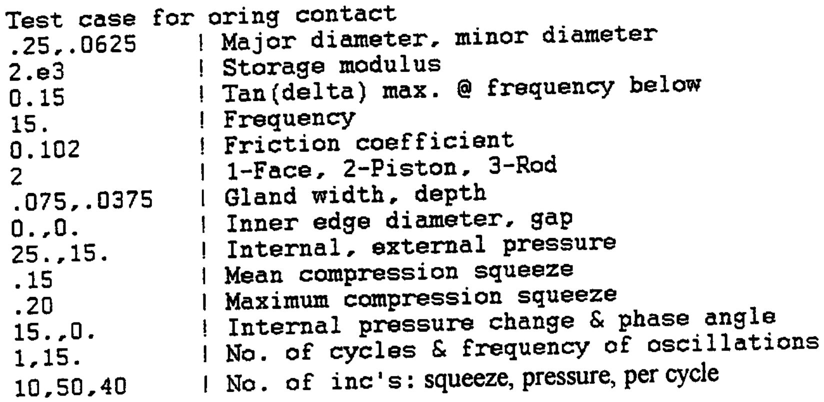


Figure 6.4 Compression Set Test Case with Varying Gap Footprint History



11/10/98 14:25:00

Figure 6.5 Compression Set Test Case with Varying Gap Normal Force History



```
Test case for oring contact
.25,.0625      ! Major diameter, minor diameter
2.e3           ! Storage modulus
0.15           ! Tan(delta) max. @ frequency below
15.            ! Frequency
0.102          ! Friction coefficient
2              ! 1-Face, 2-Piston, 3-Rod
.075,.0375     ! Gland width, depth
0.,0.          ! Inner edge diameter, gap
25.,15.        ! Internal, external pressure
.15            ! Mean compression squeeze
.20            ! Maximum compression squeeze
15.,0.         ! Internal pressure change & phase angle
1,15.          ! No. of cycles & frequency of oscillations
10,50,40       ! No. of inc's: squeeze, pressure, per cycle
```

Figure 6.6 MARC Sample User Input Data

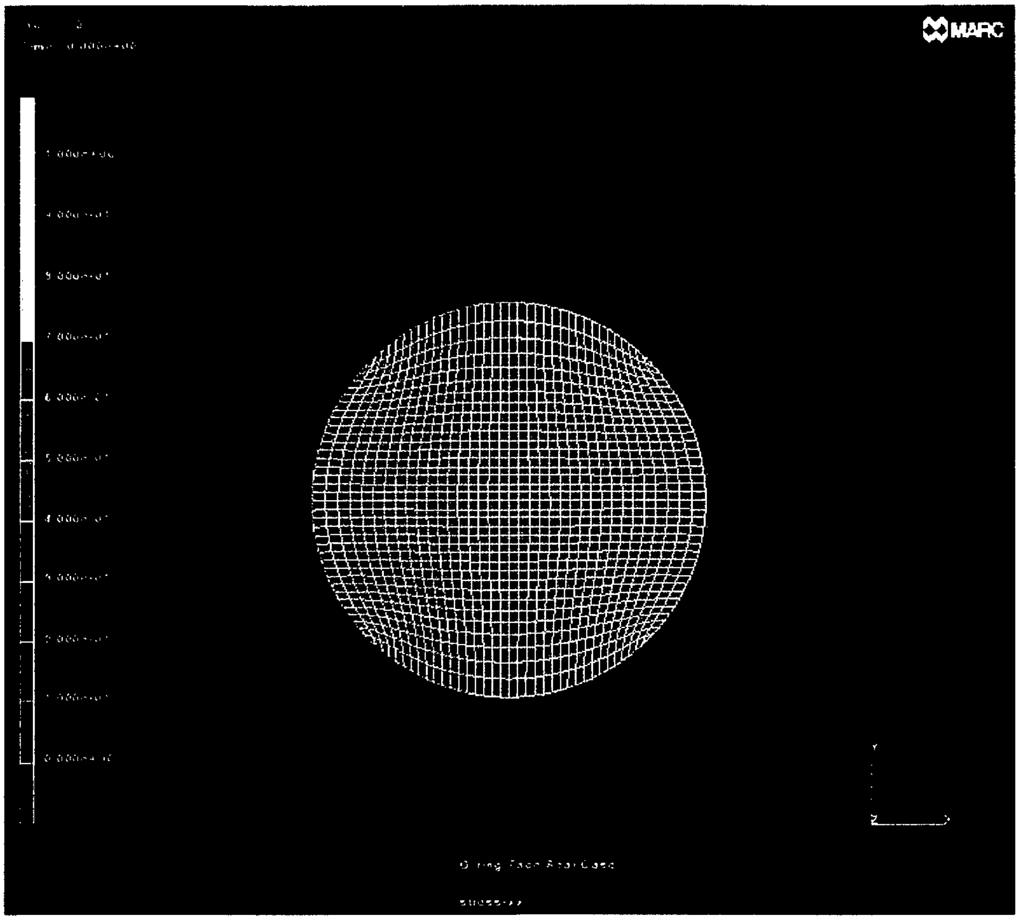


Figure 6.7 Face Seal Test Case - Geometry

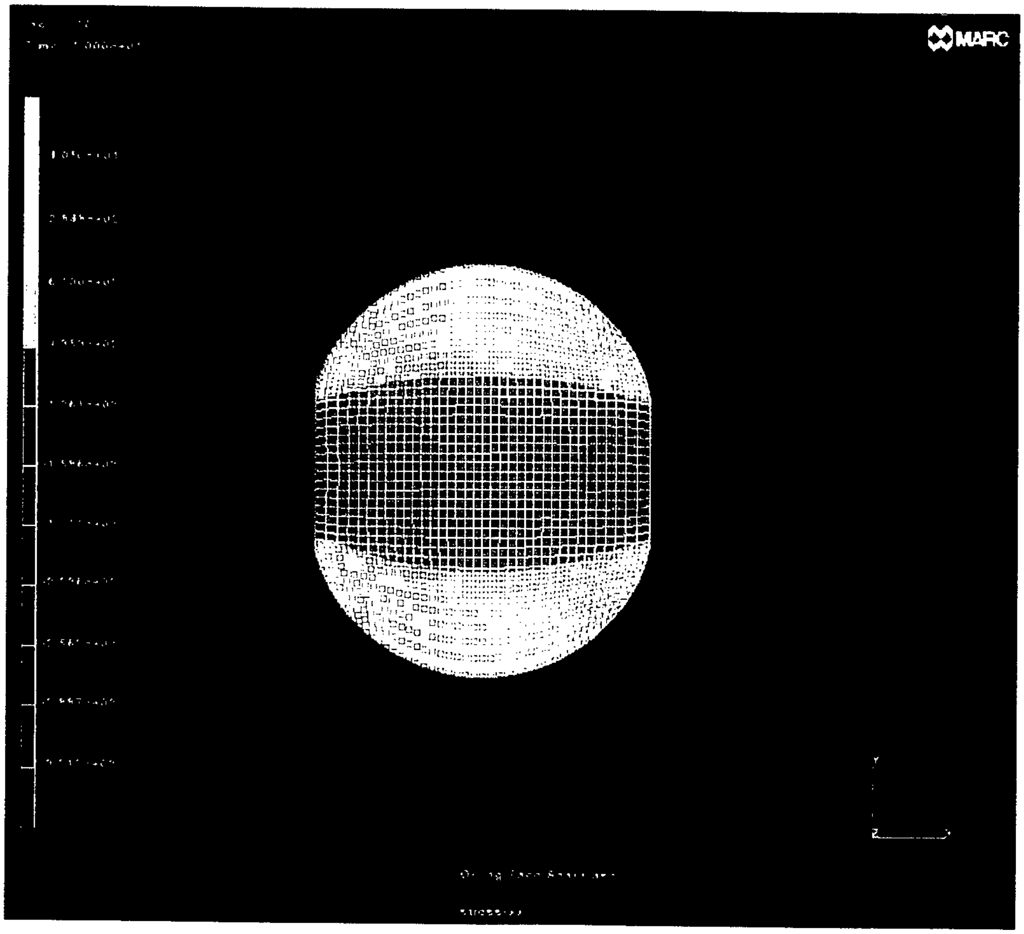


Figure 6.8 Face Seal Test Case – Intermediate Example Stress Output

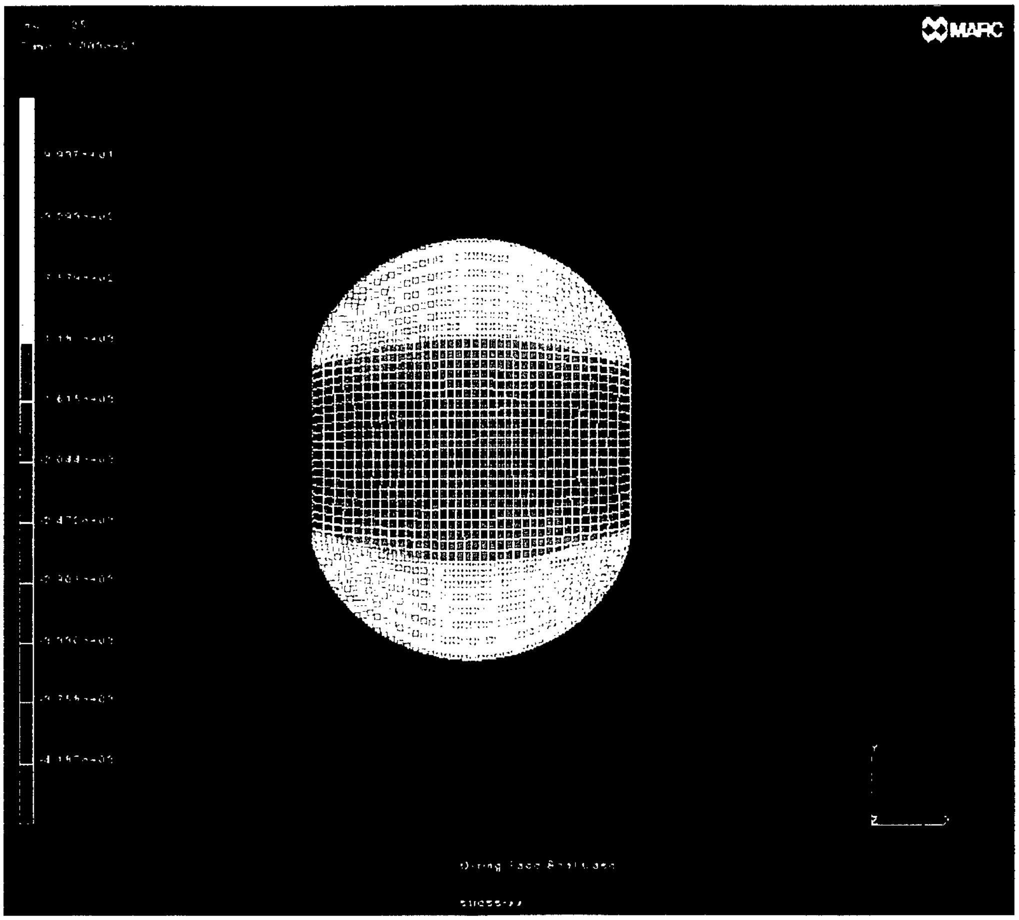


Figure 6.9 Face Seal Test Case – Final Example Stress Output

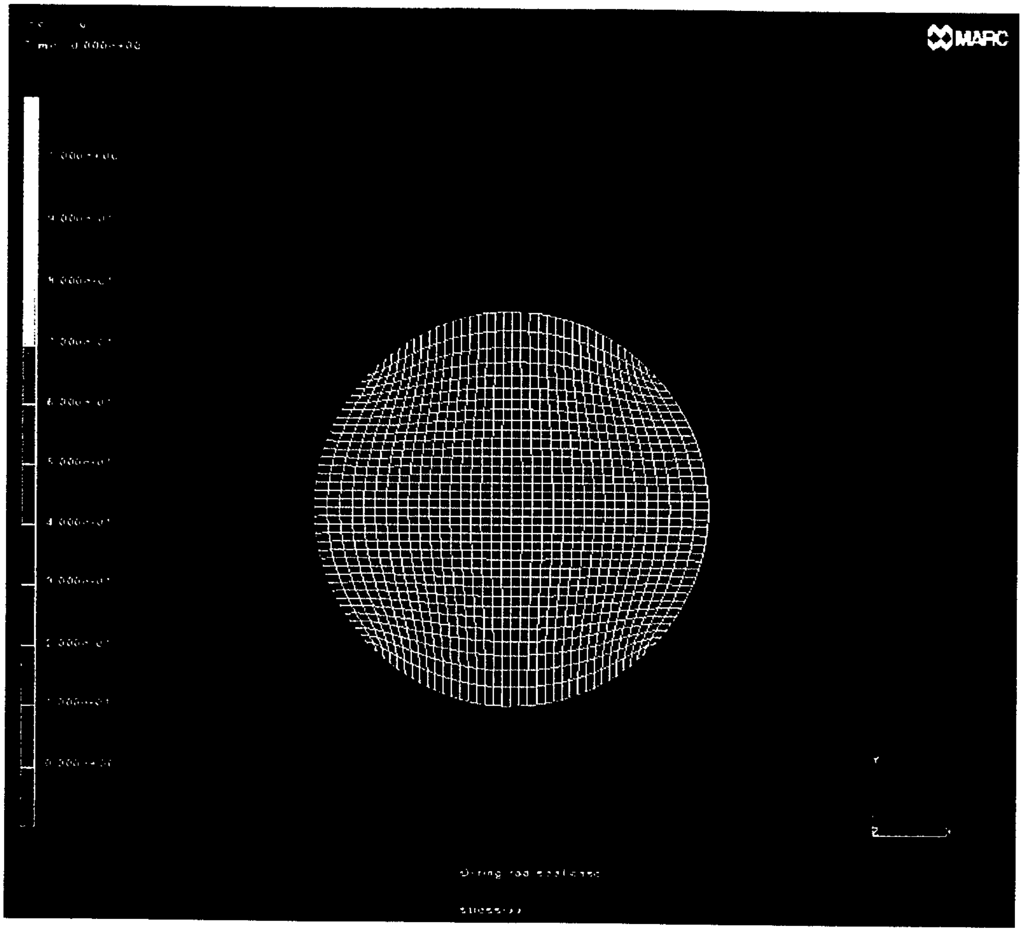


Figure 6.10 Rod Seal test Case - Geometry

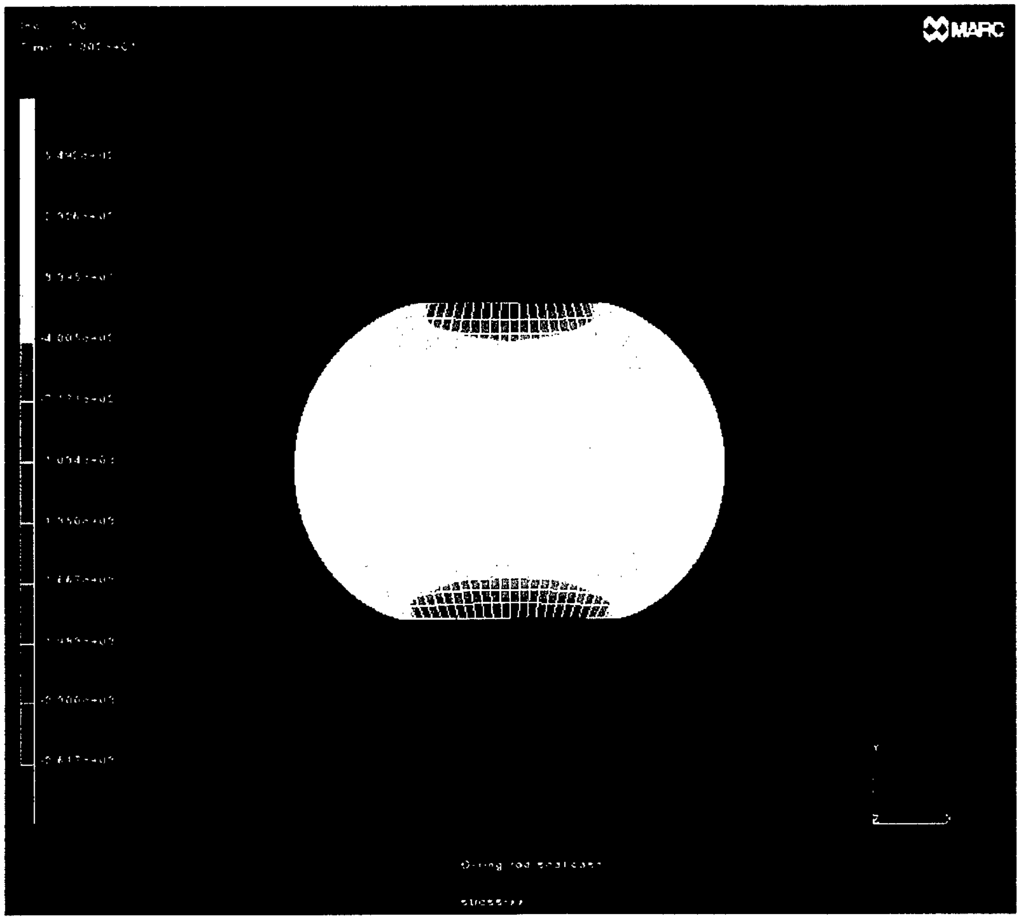


Figure 6.11 Rod Seal Test Case – Example Stress Output

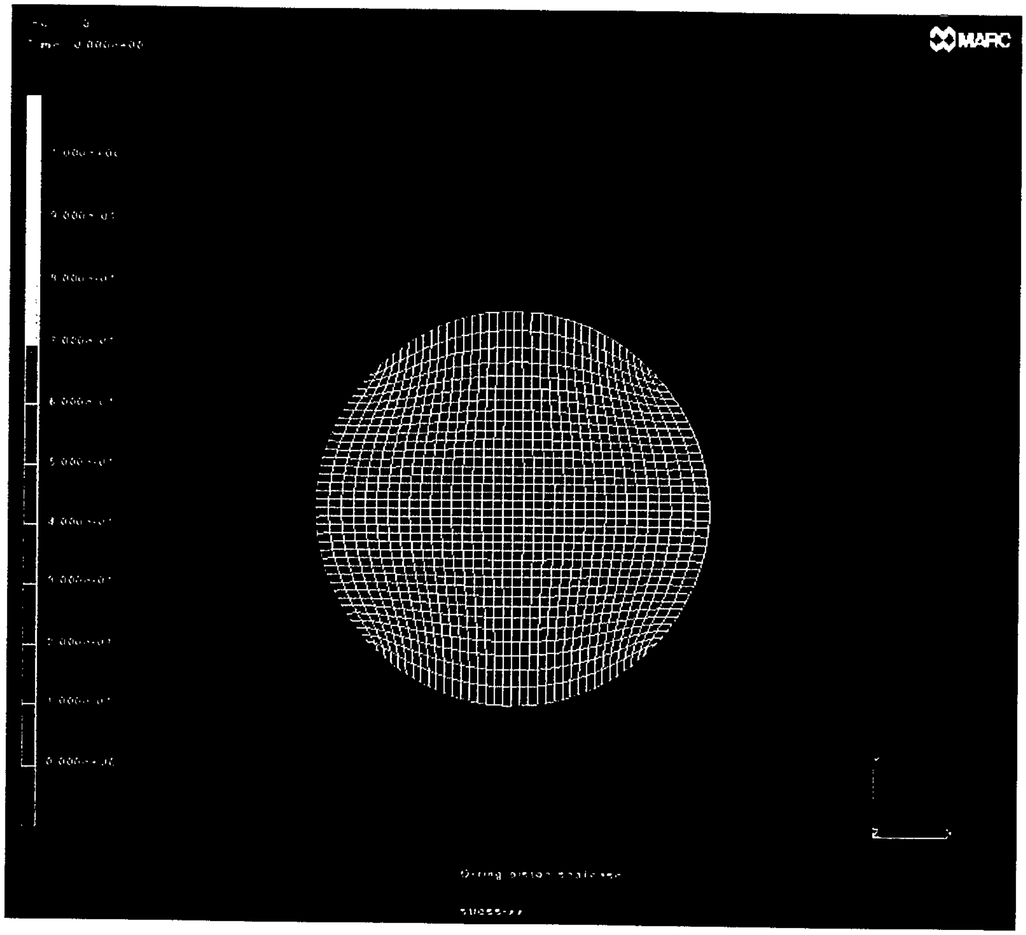


Figure 6.12 Piston Seal Test Case - Geometry

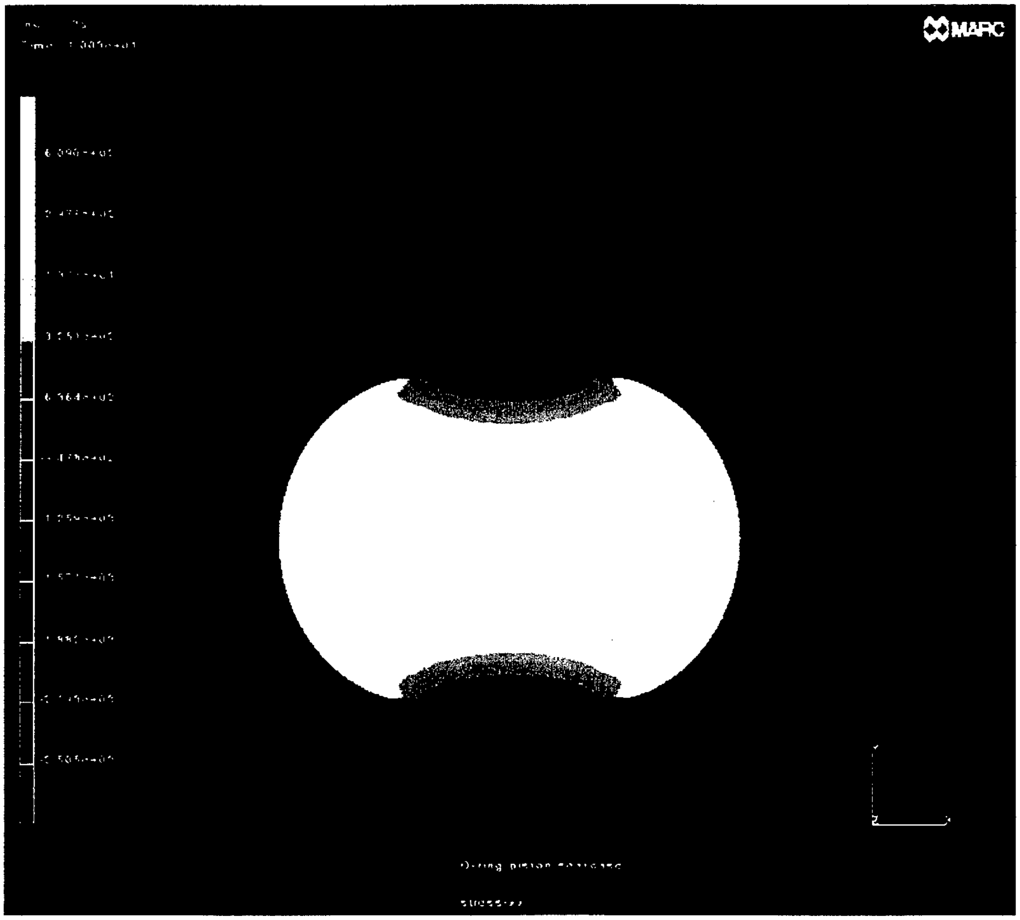


Figure 6.13 Piston Seal Test Case – Example Stress Output

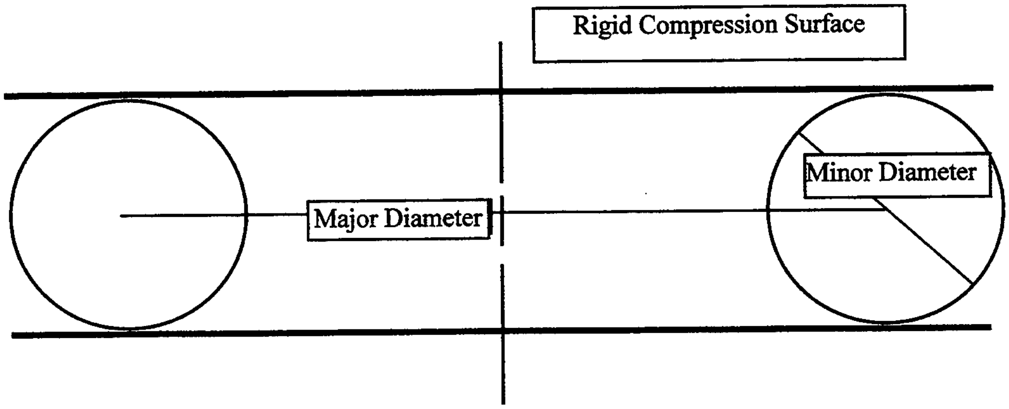


Figure 6.14: Geometry of the O-Ring Model

7.0 TEST METHODS

Determination of fundamental properties of plastic and elastic materials normally requires the use of dedicated instruments using samples with specific shapes and dimensions, such as rectangular bars and cylindrical disks. Simple tests, such as TMA (Thermal Mechanical Analysis) only determine basic characteristics, such as softening point or coefficient of expansion.

More sophisticated methods are required to extract fundamental properties, such as elastic and loss moduli. Examples of the type of instrumentation required to perform this type of testing includes the RDA II (Rheological Dynamic Analysis) from Rheometrics Corp. and the DMA 982 or 983 (Dynamic Mechanical Analyzer) from TA Instruments. A different method, described in ASTM D945, requires the use of the Yerzley mechanical oscillograph. Since the test methods required by these instruments require samples of specific shapes, these samples must be provided by removal from existing parts, where possible, or the preparation of special samples for testing. Removal of sample pieces from parts, such as, O-rings or seals, results in destruction of the part. This prevents monitoring of changes in material properties of a part due to aging or other effects.

Simple tests do exist, where the effect of impact on elasomeric samples can be measured. Examples of these methods include ASTM D1054, where a pendulum can be dropped against a rubber sample and both the penetration into the sample and the subsequent rebound height measured. In ASTM D2632, the rebound height of a ball dropped on a flat sample can be measured. For example, in ASTM D1054, since the energy of the pendulum is proportional to the verticle component of the displacement of the pendulum, it may be expressed as 1 – cos of the angle of displacement, and impact resilience, R, is readily determined from the equation:

$$R = 100 \times (1 - \cos \text{ angle of rebound}) / (1 - \cos \text{ of original angle})$$

The value R is commonly called percentage rebound. In both these methods, portions of irregular speciments could be included in the test. The energy imparted by the test system and the rebound energy can be calculated. However, a calculation method for determining basic viscoelastic properties is not possible using these methods. Additional information must be collected describing the dynamics within the test and calculation methods for extracting the basic properties would have to be developed.

7.1 New Test Methods

The material models require methods that are sensitive to subtle and continuous changes in materials properties. In general, dynamic methods, which cyclically excite the elastomer are more sensitive to changes in the material than single event testing. Methods that use cyclic excitation include torsional rheology using the RDA instrument and Dynamic Mechanical Analysis (DMA). Examples of single event testing include tensile testing, hardness testing by durometer and simple rebound testing. The significant difference between cyclic testing versus single point methods is that the material can be excited over a frequency to reach a steady state condition. Other methods of dynamic testing exist, where the sample can be excited by a single pulse and a frequency dependent response is measured. A stress relaxation test using a torsional sample in the RDA is one example. It may be possible to measure and analyze the response of an elastomer sample excited by a rebound test Proposed methods included:

1. Modified DMA or RDA testing of O-rings.
2. Dynamic analysis of rebound testing of O-rings,
3. Cyclic compressive testing of O-rings between parallel plates.

Each method would be developed and applied to new and laboratory aged O-rings. Analysis of the results would be used to determine changes in basic properties, such as storage and loss moduli.

7.2 Modified Rheometer or DMA Methods

For this type of test, the O-ring specimen would be deformed between two parallel plates using a known amount of squeeze and compressive force. One plate would oscillate the O-ring around the center axis and the axial force-displacement response measured by the second plate. Implementation of this method with existing equipment would be relatively simple; however, the analysis of the results would be difficult. The effects of slip, distortion and the interaction of a preload force normal to the direction of cyclic stress presented a serious analytical challenge. This approach was not explored further in favor of less difficult analytical models and more direct test methods, as requested by the Air Force.

7.3 Rebound Test Methods

Rebound testing is based on existing methods of exciting rubber or plastic samples by the dropping of a sphere or hammer of known mass a fixed distance onto the sample and observing the rebound response of the sample. Since the mass and height of falling element is known, the kinetic energy imparted to the test sample can be calculated. Standard tests exist for this type of testing ASTM D256, D1054, D2632 and D3574. However, these tests have limits in that the continuous response of the sample following impact is not measured. Each method was reviewed as the possible basis for a dynamic test. D256 uses a standard pendulum to measure the impact resistance of samples and D1054 measures the rebound height of a pendulum dropped on a rubber sample. D2632 and D3574, test H, measure the rebound height of a ball dropped onto a flat rubber surface or on a foam material, respectively.

Dynamic analysis of a rebound type test would require continuous measurement of the behavior of the system, to determine the position of the energy imparting component during multiple rebound events. This approach not only allows the analysis of a series of impacts and rebounds but also allows direct measurement of decay within the test system. Three approaches were considered:

1. A standard ball drop test, described in D2632, could be observed using a video recorder and a reference grid. Motion of the ball during multiple rebounds would be analyzed using the video recording. A problem with this method is the inability to control the path of the

bouncing ball. A second problem is the contact surface between the ball and the O-ring is not conducive to reproducible impacts. Alternative approaches were considered.

2. Modification of the ball drop test using a flat washer or disk in place of the ball was considered. The disk might be guided by a stiff guide wire. However, a difficulty exists for controlling the impact of the disk with the O-ring. To provide a reproducible test, the disk must land flat with each impact. This method was deferred as additional methods were considered.
3. Instrumentation of a standard pendulum to measure position was examined. Position of the arm and hammer could be measured using a mechanically coupled encoder or the position could be determined using the video camera and grid method described above.

7.3.1 Pendulum Test Method and Hardware

The advantages of the pendulum method are obvious when compared to methods using free falling spheres. Commercial test instruments, including surplus equipment at UTRC, were available for modification. The behavior of this type of pendulum is well undestood. The moving element, or hammer, travels in a fixed path, so multiple rebounds and reproducibility between experiments is possible. Incorporation of an encoder to determine hammer position would be a relatively simple task. A pendulum rebound system was selected by UTRC for further development for rebound testing of elastomer samples. The test instrument is shown in Figure 7.1. This instument is based on a single beam Wiedemann-Baldwin impact tester.

The use of a pendulum system to impart a known amount of energy is described in a number of ASTM test methods, such as D1054, D256 (Izod and Charpy methods) and D3998 (Kravitz pendulum). The advantage of a pendulum system is the potential and kinetic energy in the pendulum can be determined if the mass and geometry of the pendulum are known and the position of the pendulum is measured as a function of time. From postion-time measurements the velocity and kinetic energy can be derived. If a system rebounds against a known sample for multiple events, until complete decay occurred, the dynamics of excitation and energy dissipation can be analyzed if a satisfactory model for the behavior was developed.

The following modifications were made to the instument. The sample holder was replaced with a rigid steel anvil, designed to hold the O-ring under test against a flat surface. The face of the pendulum was fitted with a hammer of known curvature. The pendulum weight distribution was determined. The fixed release mechanism at the top of the pendulum was attached to a movable arm, providing release points from 15 to 150 degrees in 15 degree steps. The axial shaft from which the pendulum swings was fitted with a low mechanical resistance, linear potentiometer. The potentiometer was connected to a regulated 10 V power supply. The output of the potentiometer was connected to a zero gain buffer amplifier with high input impedance. The position of the pendulum beam was calibrated vs. output voltage. This output was connected to a multi-channel A-D data acquistion unit running on a MS-DOS/Windows based computer. Details are shown in Figure 7.2a and 7.2b.

7.3.2 Pendulum Test Software

The pendulum tests are transient and highly nonlinear. The nonlinearities are due not only to the contact during the impact of the hammer with the O-ring but also are introduced by the large amplitude motion of the pendulum. Accurate models of the test would require simulations using finite element codes. This would be well beyond the capability of the ordinary PC's envisioned for use with the pendulum impact tester. Hence, a simple method for interpreting the data from a pendulum tester needs to be developed.

Several inputs are required to reduce the data from a typical pendulum impact test. The data can be placed into three categories: the characteristics of the pendulum, the geometry of the O-ring, and the location of the pendulum as a function of time. The pendulum can be described by: the inertia, *I;* the length from the pendulum pivot to the impact point, *L*; the length from the center of gravity to the pivot, *r*; and the chord impact length, c_l. The chord length, c_l, divided by the circumferential length is the fractional length over which the impact occurs. If the hammer were larger than the entire O-ring the ratio would be one. Usually the ratio is much smaller than one. The geometry of the O-ring includes the major diameter (inside diameter plus half the cross-section diameter), *D*, and the minor (or cross-section) diameter, *d.* The input data from the test includes the angular position of the hammer as a function of time, collected at a sample rate high enough to determine the time the hammer is in contact with the O-ring.

The center of mass of the pendulum arm can be found by balancing. The inertia can be found by the measuring the small amplitude frequency, *f*, and then using:

$$I = \frac{mgr}{4\pi^2 f^2} \qquad (24)$$

The radius of gyration, *k*, can be used in place of the inertia, if the mass is known, and can be found from the definition:

$$k^2 = I / m \qquad (25)$$

The motion of the pendulum can be adequately described by considering the conservation of energy. The energy, *E*, is given by:

$$E = \frac{1}{2} I \dot{\theta}^2 + mgr(1 - \cos\theta) \qquad (26)$$

where θ is the angular position, *( )=d/dt* is the derivative with respect to time, and g is the acceleration of gravity. The energy can be found for each impact by determining the maximum angle before the impact (where the angular velocity vanishes.) The energy is then given by:

$$E = mgr(1 - \cos\theta_{\max}) \qquad (27)$$

The angular velocity at the beginning of the impact is a maximum and can be found by setting θ equal to zero in equation (26) as:

$$\dot{\theta}_{\max} = \sqrt{2E/I} \tag{28}$$

The impact will now be considered as a mass moving at some initial speed into a linear spring and mass in parallel. This is only a rough approximation and later work can refine the description of the impact. The ratio of the velocity before the impact divided by velocity after the impact is related to the fraction of critical damping, ζ, by:

$$\zeta = \ln(\dot{\theta}_{before} / \dot{\theta}_{after}) \tag{29}$$

The time of impact, *dt*, (which can be extracted from the data if the sampling rate is sufficiently high) is one half cycle and can be used to find the natural frequency, ω, (in radians per second) as in:

$$\omega = \pi / dt \tag{30}$$

The effective stiffness, *K*, can now be found from:

$$K = m(\omega k / L)^2 \tag{31}$$

while the effective damping constant, *c*, is:

$$c = 2\zeta m\omega(k / L)^2 \tag{32}$$

The *tanδ* for the O-ring material can now be found by dividing force constant by the spring force:

$$\tan\delta = \frac{\omega c}{K} \tag{33}$$

The problem now is to translate the effective spring constant into a shear modulus. The shear modulus will be proportional to the effective spring constant and inversely proportional to the distance over which the impact occurs (i.e. the chord length, c_l) or:

$$G^* = \alpha K / c_l \tag{34}$$

where

$$G^* = \sqrt{G'^2 + G''^2} \tag{35}$$

From a simulation of an O-ring squeezed between two parallel surfaces, an approximate value for the constant α is:

$$\alpha = 3.70 \tag{36}$$

Equations (34) and (36) are only an estimate, and a more accurate representation is required. The storage and loss moduli can now be found from:

$$G' = G^{*}\cos\delta \tag{37}$$

and

$$G'' = G^{*}\sin\delta \tag{38}$$

A more accurate interpretation can be found by determining the energy as the area under the load deflection curve for the O-ring squeezed between two parallel surfaces as a function of the deflection. Since the energy input is known and the deflection can be found from accurate experimental data, the effective modulus could be found. Unfortunately, with the current pendulum test hardware, it was not possible to determine the deflection to a sufficient accuracy. Improvement of accuracy and resolution of the pendulum hammer position and deflection would be required to provide a better interpretation.

A description of the software used to interpret the pendulum test data is included in Appendix E, along with a complete listing.

7.3.3 Discussion of Pendulum Rebound Test Results

Master data describing the history and dynamic testing results for O-rings is shown in Table 7.1 for the MIL-R-25988 fluorosilicone O-rings and in Table 7.2 for the nitrile O-rings. The aging history, compression set data, dynamic parallel plate testing and dynamic pendulum testing results are all reported in these tables. Testing was limited to the larger, AS 568 size -214, O-rings. Significant breakage occurred with the smaller, size -007 samples, limiting the number of specimens available for testing. Due to limited resources, a decision was made to concentrate on the developing and calibrating the larger pendulum hammer that would be used with the –214 O-rings. Calculations were made to determine the fundamental frequency (Rebound_Freq_n)), the storage modulus, G' (Gpn) and loss modulus, G" (Gppn) of the first three rebounds (n) for each measured sample. The mean values of three rebound tests per sample are shown in the table.

For each manufacturer, preliminary tests were run on new O-rings. Tests were run on dry specimens, followed by tests on new O-rings which had been wet with test fluid. JP-8 was used for the fluorosilicone seals and hydraulic oil for the nitrile seals. Dry samples were run only for reference purposes. Since all the aged O-rings were stored and tested wet, the new, wet samples provided the baseline reference to determine property shifts. A comparison of the storage and loss moduli for new and aged O-rings is shown in Figures 7.3 and 7.4. This data has been grouped by manufacturer.

Examination of the data for the fluorosilicone O-rings indicates that significant scatter exists across the data for each manufacturer's seals. No significant trends in property shifts were seen over the course of the aging experiments. Either the property changes due to aging were insignificant or the test method was insensitive to property shifts. However, results were generally reproducible for individual samples. This suggests that the method, within inherent mechanical and instrumental limits, is sound. As mentioned in Section 7.3.2, an improvement of accuracy and resolution of the instrumentation would be required to provide a better interpretation.

Examination of the results of the rebound testing of the nitrile O-ring shows similar results. Although the range of the data (moduli in MPa) is reasonable for the type of material under test, no trends are evident. Either the aging method is insufficient to generate significant differences in the samples, or the test method is limited in sensitivity.

7.4 Cyclic Parallel Plate Test

Under actual use conditions, O-ring seals can be considered to be trapped between two parallel plates with some nominal level of compressive strain. Under actual use, the seal is trapped in a groove, called a gland. Typical O-ring seal configurations were previously shown in Figure 6.1. Under these conditions, the stress induced in the seal material provides a nominal sealing force. Under actual use, however, differential pressure exerted on the seal creates a more complex situation which deforms the seal material. The complex stress induced in the seal that is trapped in the gland structure provides the required sealing force, if the seal behaves as designed.

However, it may be possible to determine changes of relative sealing force between samples with different aging histories and between different materials, by cyclically exciting the compressed seals and analyzing the stress-strain response.

7.4.1 Parallel Plate Test Hardware

In practice, the seals could be held between two parallel plates and excited by cyclically varying the spacing between the plates. For testing of the seals, this was accomplished by use of a pair of circular plates which were installed in a high cycle rate MTS 820 servohydraulic tester. The test samples were compressed to 80% of their relaxed height and allowed to relax for 2-3 minutes. Each sample was then oscillated between 70-80% of the original seal height at frequencies from 1 Hz to 60 Hz and the force-displacement response was measured.

In the experimental test setup, the MTS servohydraulic tester was driven by an external sinusoidal source and the relative position of the plates determined using a simple LVdt position sensor. The output of the sensor was used to provide feedback control of the servohydraulic hardware and also used as one of the measured outputs. A load cell was placed in series with the static (reference) plate to measure force output. The plates were ported to relieve gas pressure from the cavity between the two plates and the seal. Based on prior testing of various types of elastomeric specimens, this configuration generally gives good results in the 0.1-60 Hz range with small displacements, typically under 0.100 inches. (2.54 mm) The parallel plate tester is shown in Figure 7.5.

To acquire data, two methods were used. Initially, due to limitations of available digital data acquisition hardware, a Techtronix Model DSA 602A recording oscilloscope was used for data acquisition. Digital data acquisition hardware available at the start of testing was limited to less than 100 Hz per channel and contamination of data was a concern. Samples were oscillated between 70-80% of relaxed height at 1, 5, 10, 20 40 and 60 Hz. The force and displacement output from each test was printed and set aside for analysis. This method provided an adequate method for recording data. However, it made analysis difficult due to the manual method of data analysis. Later samples, (for example, the nitrile O-ring seals) were tested using similar methods, and data acquisition was provided using Strawberry Tree data acquisition hardware and a 486 computer running Workbench for PC software. This data acquisition method provided a sufficient bandwidth of 500-2000 Hz per channel depending on hardware configuration.

The parallel plate test is not linear because of the area changes and, hence, is difficult to interpret. The most accurate interpretation would be to use the MARC or ANSYS workstation software to simulate the test. This would automatically include the nonlinear effects.

7.4.2 Parallel Plate Test Software

The results from the cyclic parallel plate tests consist of the maximum and minimum loads for each cycle. The input conditions include: the mean squeeze (i.e., the mean displacement across the cross-section diameter divides by the diameter) and the change in the squeeze. Typically the tests were performed at mean of 25 percent with a variation of 5 percent above and below the mean.

A relatively simple method for interpreting the data would be to take the contact area as the contact area at the mean squeeze and the induced strain as the strain at the center of the O-ring cross-section. A MARC finite element model for an elastic O-ring with a typical ratio of major-to-minor diameters was used to find the contact area and the strain as a function of the squeeze. The area ratio is defined as:

$$A_R = \frac{A_c}{\pi D d} \qquad (39)$$

where A_c is the contact area, D is the major diameter., and d is the minor diameter. A least squares fit to the results of the simulation (for all but the first point) is shown in Figure 7.6a. The least squares fit for the area ratio is:

$$A_R = 1.823S + 0.0958 \qquad (40)$$

where S is the squeeze. In a similar manner the strain is given by:

$$\varepsilon = 1.270S - 0.040 \qquad (41)$$

and is shown in Figure 7.6b. The strain range can now be found from:

$$\varepsilon_{p-p} = \varepsilon_{\max} - \varepsilon_{\min} \qquad (42)$$

where the peak-to-peak squeeze is twice the maximum minus the mean squeeze.

The effective modulus can now be found from:

$$G^* = \frac{F_{\max} - F_{\min}}{3\pi D A_R \varepsilon_{p-p}} \qquad (43)$$

where use has been made of equation (3).

The phase angle between the median force and the median displacement times can be used to find δ (recall that the ratio of G' to G'' is $tan\delta$) as:

$$\delta = \pi f(t_{up} + t_{down}) \qquad (44)$$

where t_{up} is the time the median force leads the mean squeeze, and t_{down} is the lead time going down. The storage and loss modulus can be now be found from equations (37) and (38). Of course,

$$\tan\delta = G'/G'' \qquad (45)$$

The above data reduction can be readily completed using a PC spreadsheet.

7.4.3 Discussion of Cyclic Parallel Plate Test Results

Results from analysis of the Parallel Plate Cyclic compressive testing for fluorosilicone and nitrile O-ring seals are shown in Tables 7.1 and 7.2. Preliminary baseline tests were made using new O-ring samples from each manufacturer in both the dry state and with each O-ring wet with the appropriate fluid. The new, wet samples were used for a reference baseline for each manufacturer. The storage modulus, G' (Gp_n) and loss modulus, G" (Gpp_n) are reported in units of MPa at four frequencies in Tables 7.1 and 7.2. As shown by the test data, the results mirror the results from the rebound pendulum methods. Although the results generally gave acceptable results in terms of the absolute range of material properties, the scatter and the lack of relative change within each data set between new and aged samples indicates the aging method is insufficient to generate significant differences in the samples and the test method limited in sensitivity. A comparison of the storage and loss moduli for new and aged O-rings is shown in Figures 7.6 and 7.7. This data has been grouped by manufacturer.

7.5 Compression Set Testing

When elastomeric materials are subjected to continuous, long-term compression, creep and relaxation can occur within the sample, resulting in permanent deformation of the sample. This phenomenon is called compression set and was previously discussed in Section 6. The definition of compression set is shown in Figure 2.2.

7.5.1 Compression Set Test Hardware

As described in Section 5.2, prior to dynamic testing, the O-ring seals were subjected to an aging process using simulated operating conditions. During aging, the seals were compressed in fixtures similar to actual operating hardware. Examples of the test fixtures are shown in Figures 5.1 and 5.2. During the aging process, the weight change and compression set were determined.

Prior to installation in the aging fixtures, the weight of each O-ring sample was recorded. The small diameter of each specimen was measured manually at 6 points using a digital micrometer. The technician responsible for measurement used an optical magnifier of approximately 10X to determine that negligible deformation of each sample occurred during the measurement method. Three points were collected along the short axis of the O-ring (thickness) and 3 points were collected across the small diameter along the long axis (width) Following each aging cycle, the samples were recovered, weighed and measured within 1-2 hours of removal from the aging fixture. The results were averaged and reported in Tables 7.1 and 7.2.

7.5.2 Reduction of Compression Set Test Data and Comparison of Results

The initial and final width and height of each O-ring tested was measured and tabulated in a spreadsheet. Three measurements were made and averaged and the initial and final mass were entered in a separate spreadsheet. An example is shown in Figure 7.9. As can be seen from Figure 7.9, it is difficult to note any correspondence between supplier and the permanent changes due to aging. Figures 7.9 through 7.17 summarize this data. Note that mass measurements were not made for all the specimens. Two aging times were considered: 168 hours or 7 days, and 504 hours or 21 days. The nitrile seals were tested at 107°C and 135°C, while the fluorosilicone seals were tested at 121°C and 149°C.

In general, the width permanently decreased and the height increased for the nitrile seals, while for the fluorosilicone seals both the width and the height permanently increased. The nitrile seals generally exhibited a mass increase, as did the low temperature fluorosilicone tests. However, the high temperature fluorosilicone seals exhibited a mass decrease. This indicates that there may have been a change in the fluorosilicone seals between 121 and 149°C. Table 7.3 averages compression set data over all suppliers.

In general, the changes should increase in absolute value with temperature and time. Since the pressure dominates over the squeeze [the squeeze can relax over time, but the loads cannot as shown in equation [(23)], the width changes must be considered over the height or mass changes. Yet the fluorosilicone shows no discernable trends. The height changes should be steady if relaxation has dominated. Again the fluorosilicone does not show a proper trend. The nitrile seals were consistent in both the height and width.

The reduction of the data can be illustrated by considering the nitrile seals. The 10°C increase in temperature for the 168 hour tests increased the width change by 2.4 times, while the increase in temperature at 504 hours increased the width change by 1.4 times. Hence, take the increase to be about 2 times for the 10°C change. The change in the width will vary as *exp(-Q/RT)*. For this case then, <u>*Q/R*</u> is about 7000 R. Taking a creep law that includes time hardening

$$\dot{\varepsilon} = A_0 e^{-Q/RT} p^n t^m = A e^{-Q/RT} t^m \qquad (46)$$

where A_0, n and m are constants. For our case, the pressure is a constant and the term $A_0 p^n$ is a constant A. The constants can be evaluated by noting that over a change in time the permanent changes again differed by a factor of two. This makes m about 0.6. Obviously, the reduced data is not accurate, and provides only an estimate of the permanent changes at other temperatures and times. Unfortunately, only one pressure was applied for each seal, and so an estimate for the stress exponent cannot be found. Time hardening is also not a good approximation of creep. In general, the compression set data did not provide sufficiently accurate data to provide a procedure for determining the life. It appears that a better estimate for the long time behavior can be discerned from the $tan\delta$, or possibly from the G' and the G'' data.

7.6 Comparison of Results

As was originally believed, the compression set data proved to be a poor prediction of changes in physical properties of O-ring seals, and, therefore, is not a good method for predicting seal life.

However, the data scatter and lack of trends observed for both the pendulum rebound test and for the cyclic compression testing was unexpected. In both cases, the level of scatter was found to be as severe as the scatter for compression set results. No trends were determined for either method to correlate the change in physical properties of each manufacturer's seals as a result of the aging process. All tests, however, were carried out with wet O-rings, and the plasticization of the seals by the fluid may obscure slight material changes.

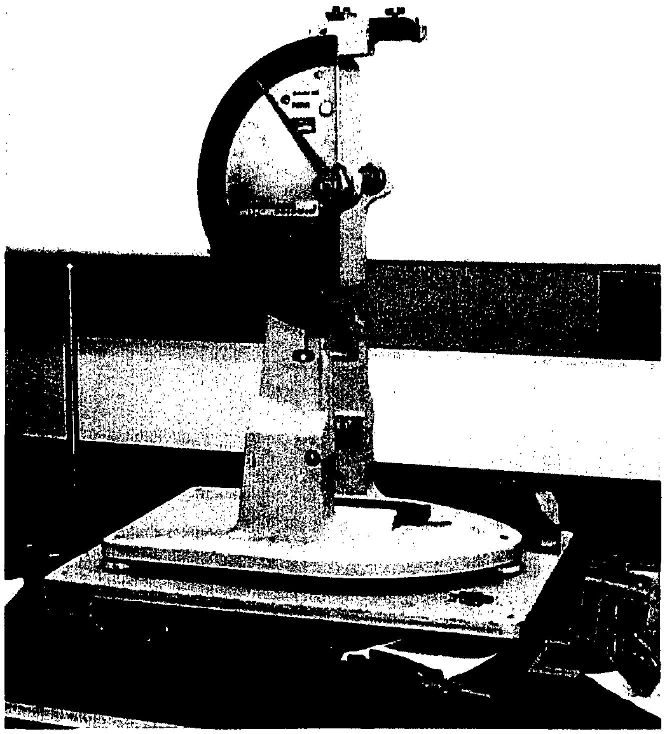


Figure 7.1 Wiedemann-Baldwin Pendulum Impact Tester

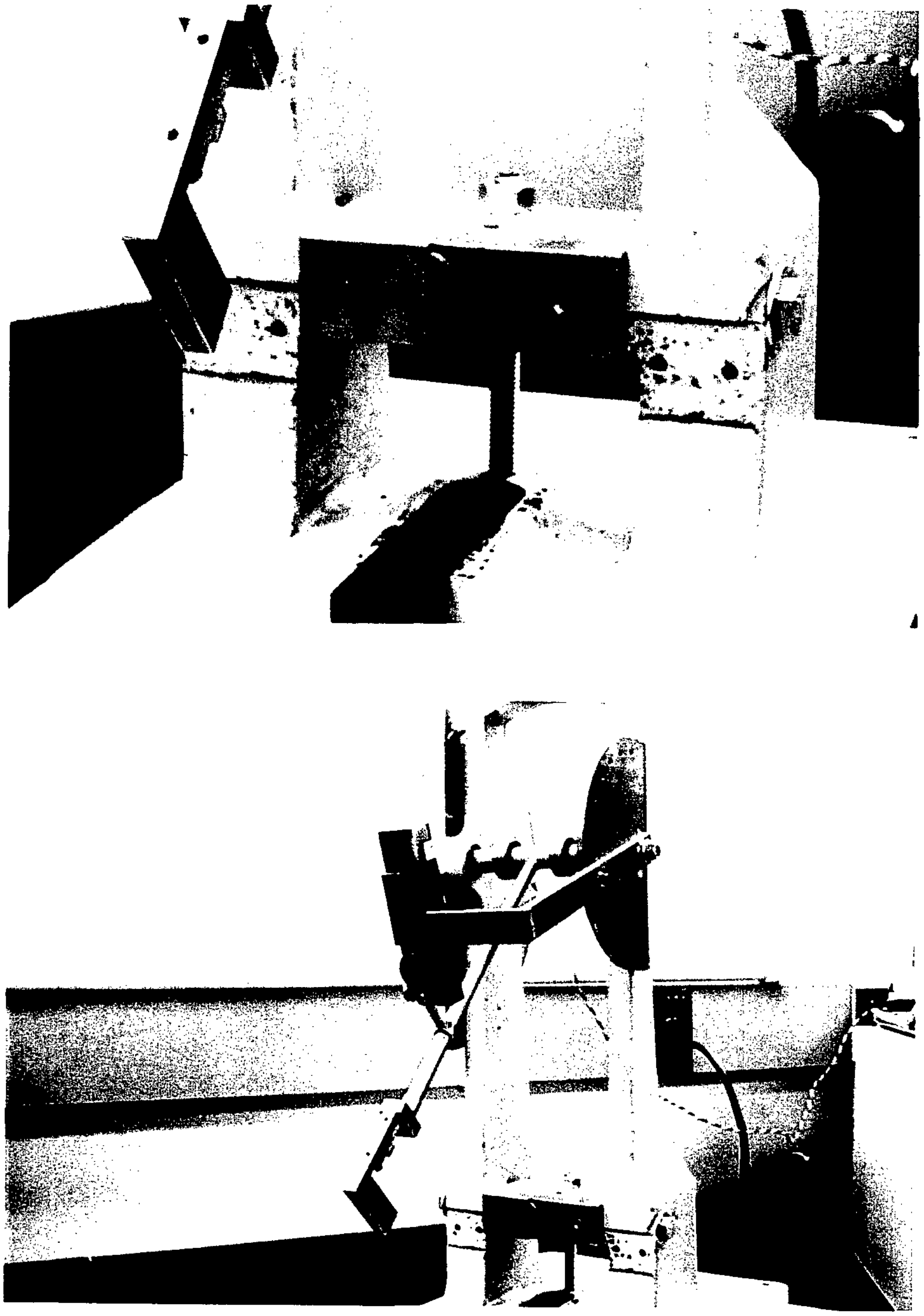


Figure 7.2a Modified Pendulum Rebound Tester

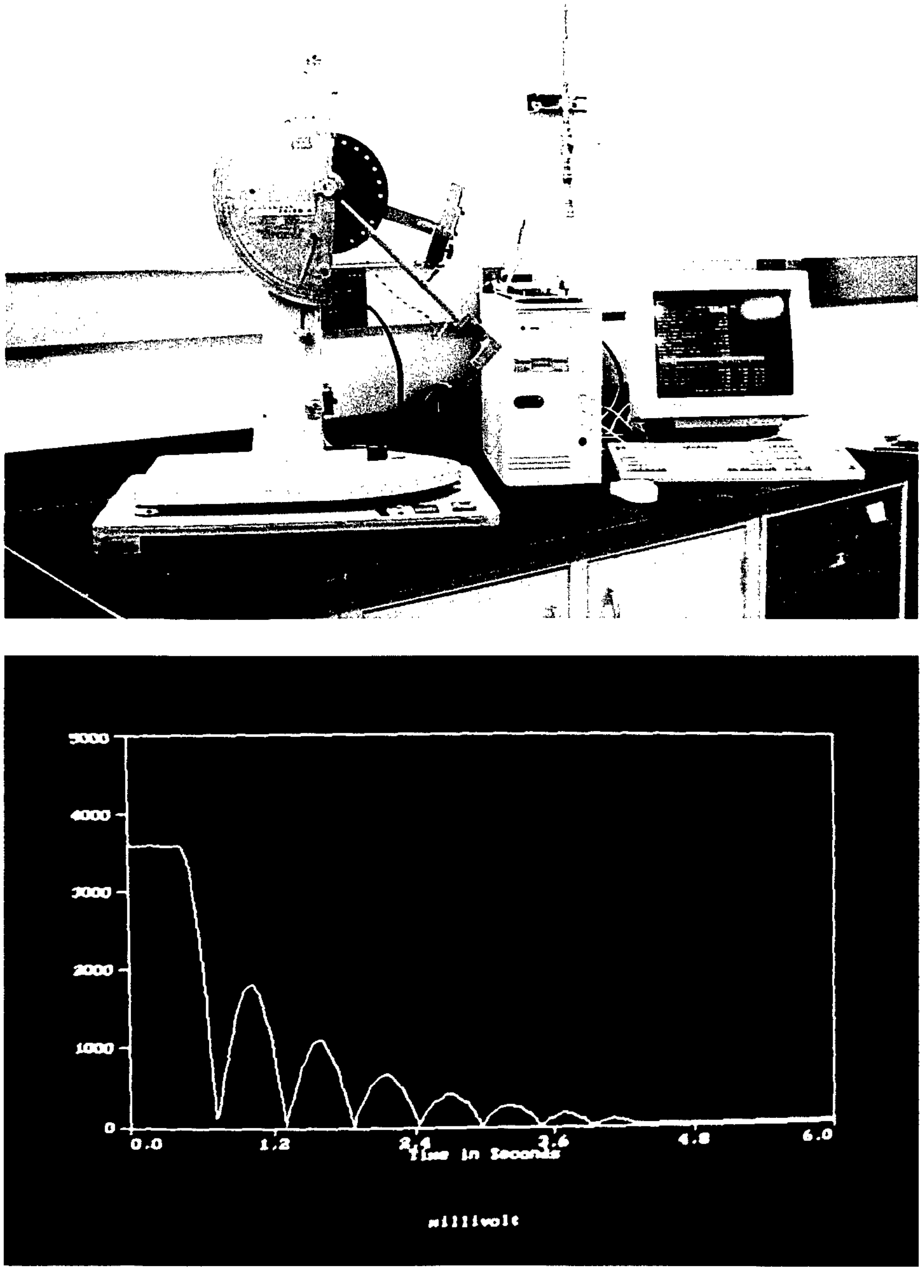


Figure 7.2b Modified Pendulum Rebound Tester

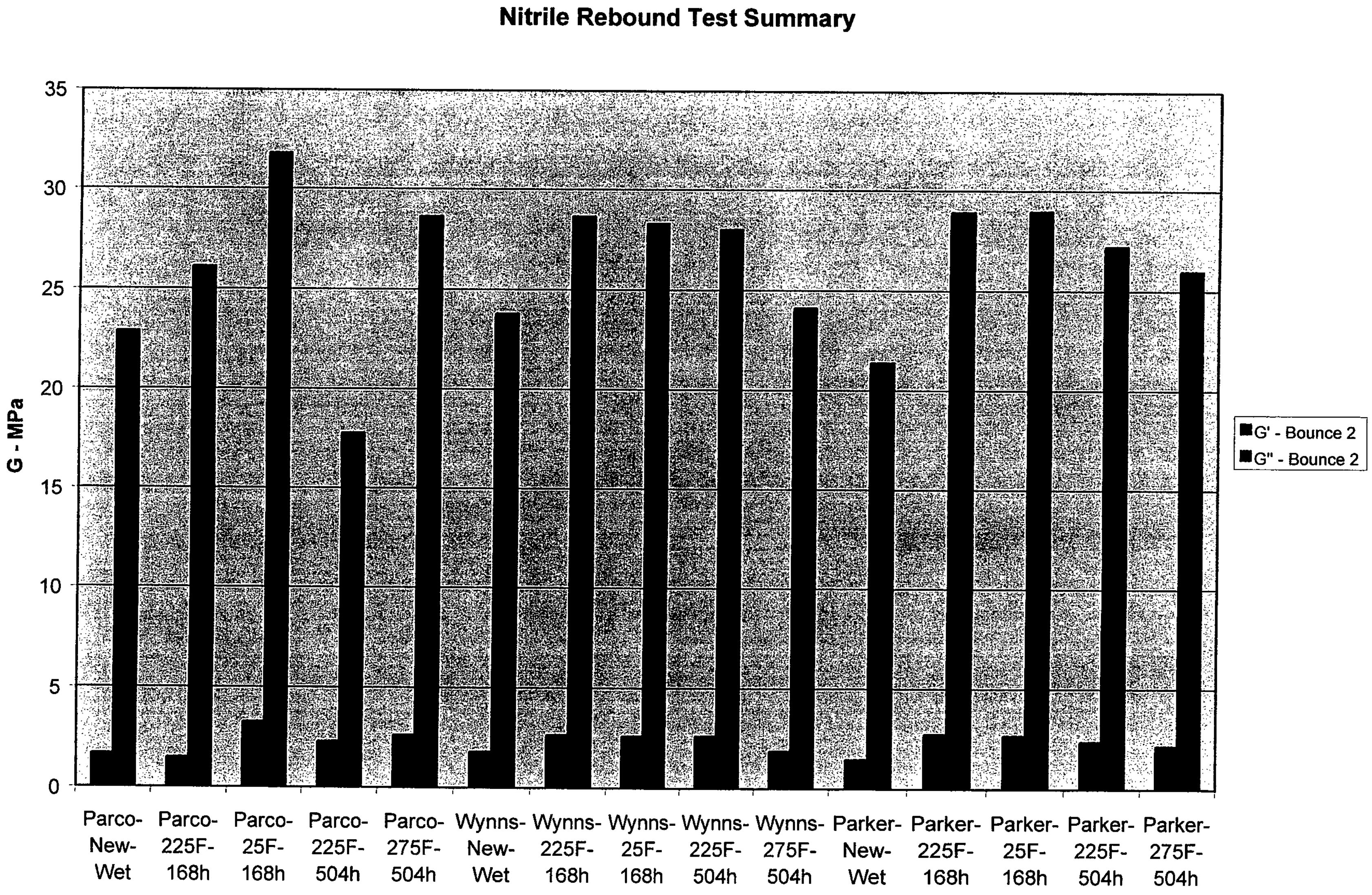


Figure 7.3: Summary of Nitrile Rebound Testing, 2^{nd} Bounce Data

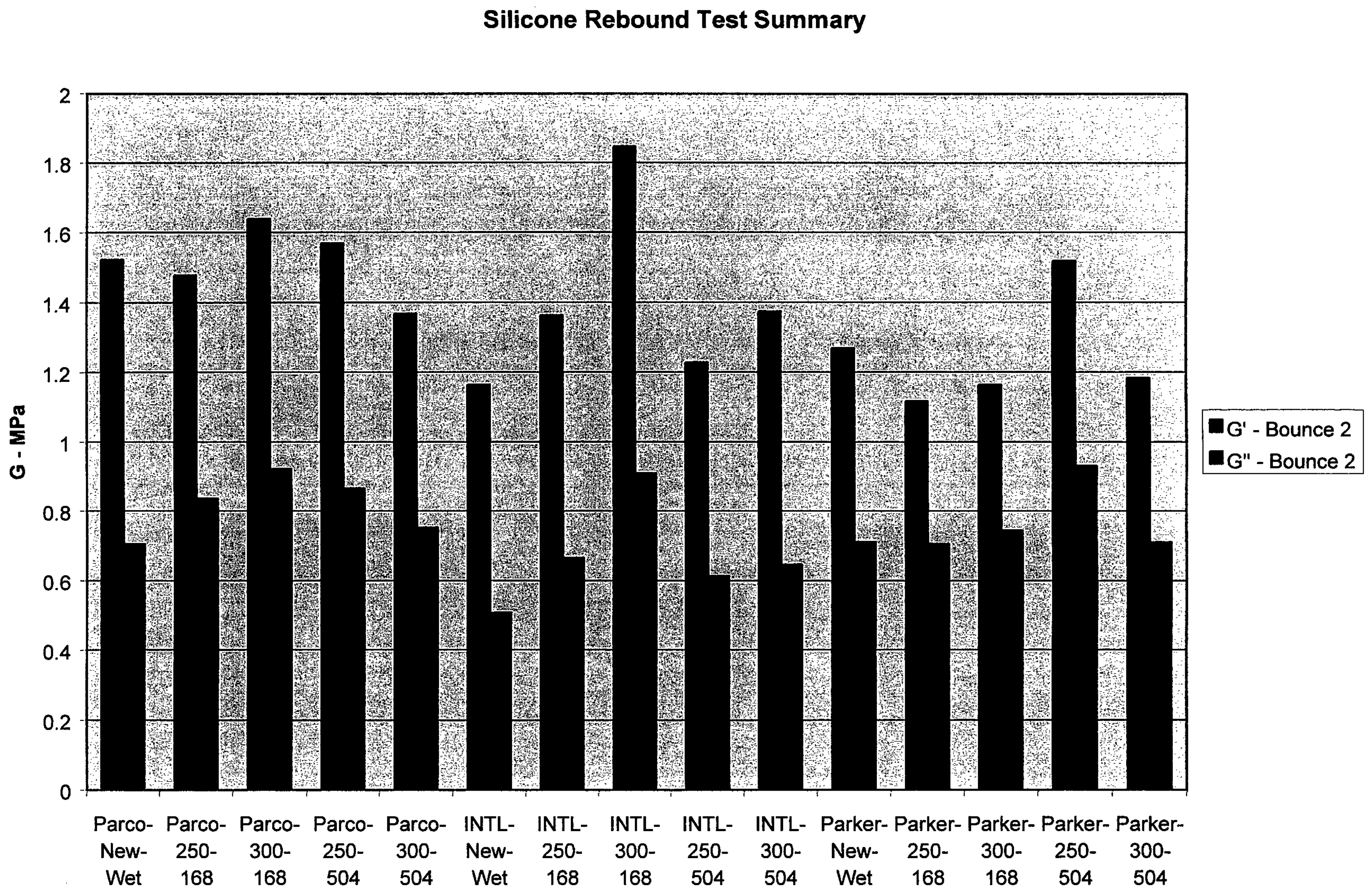


Figure 7.4: Summary of Silicone Rebound Testing, 2^{nd} Bounce Data

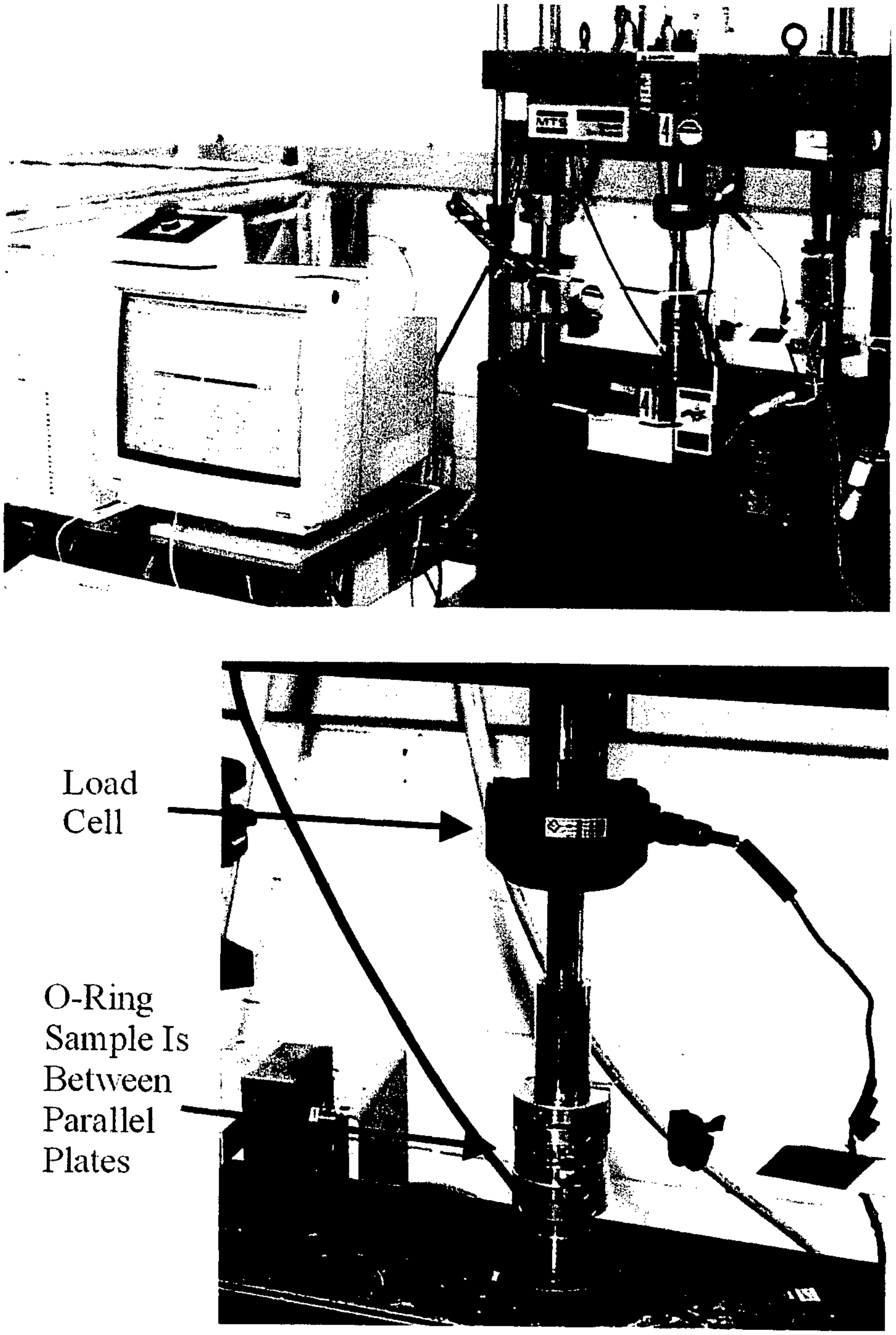


Fig 7.5 Cyclic Compression Parallel Plate Tester

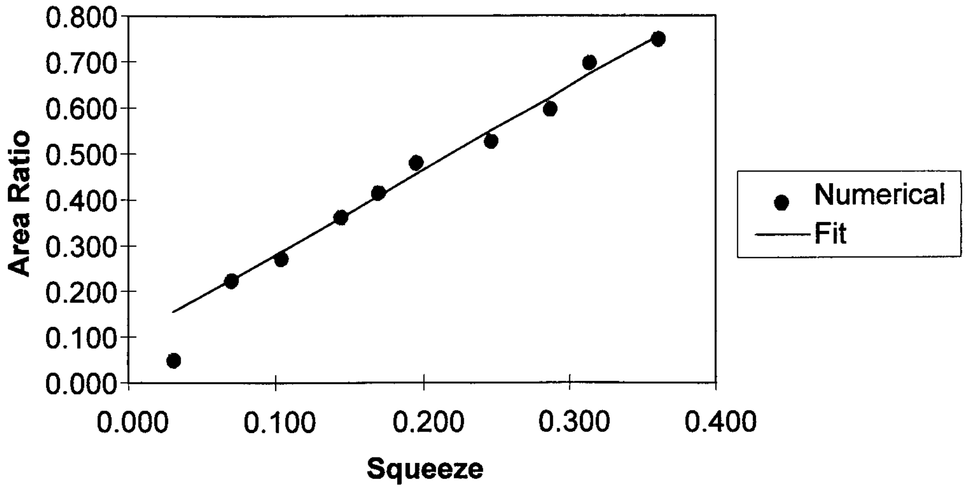


Fig. 7.6a Least squares Fit for Area Ratio

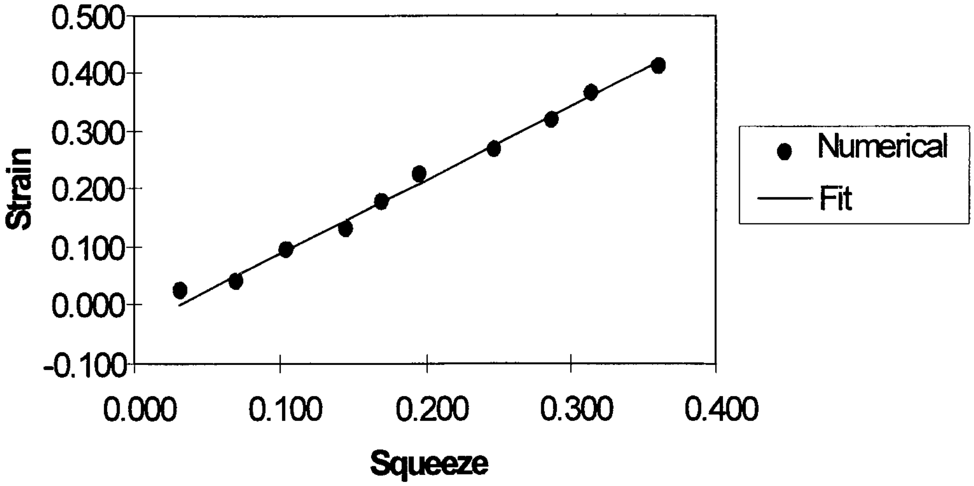


Fig. 7.6b Least Squares Strain Fit

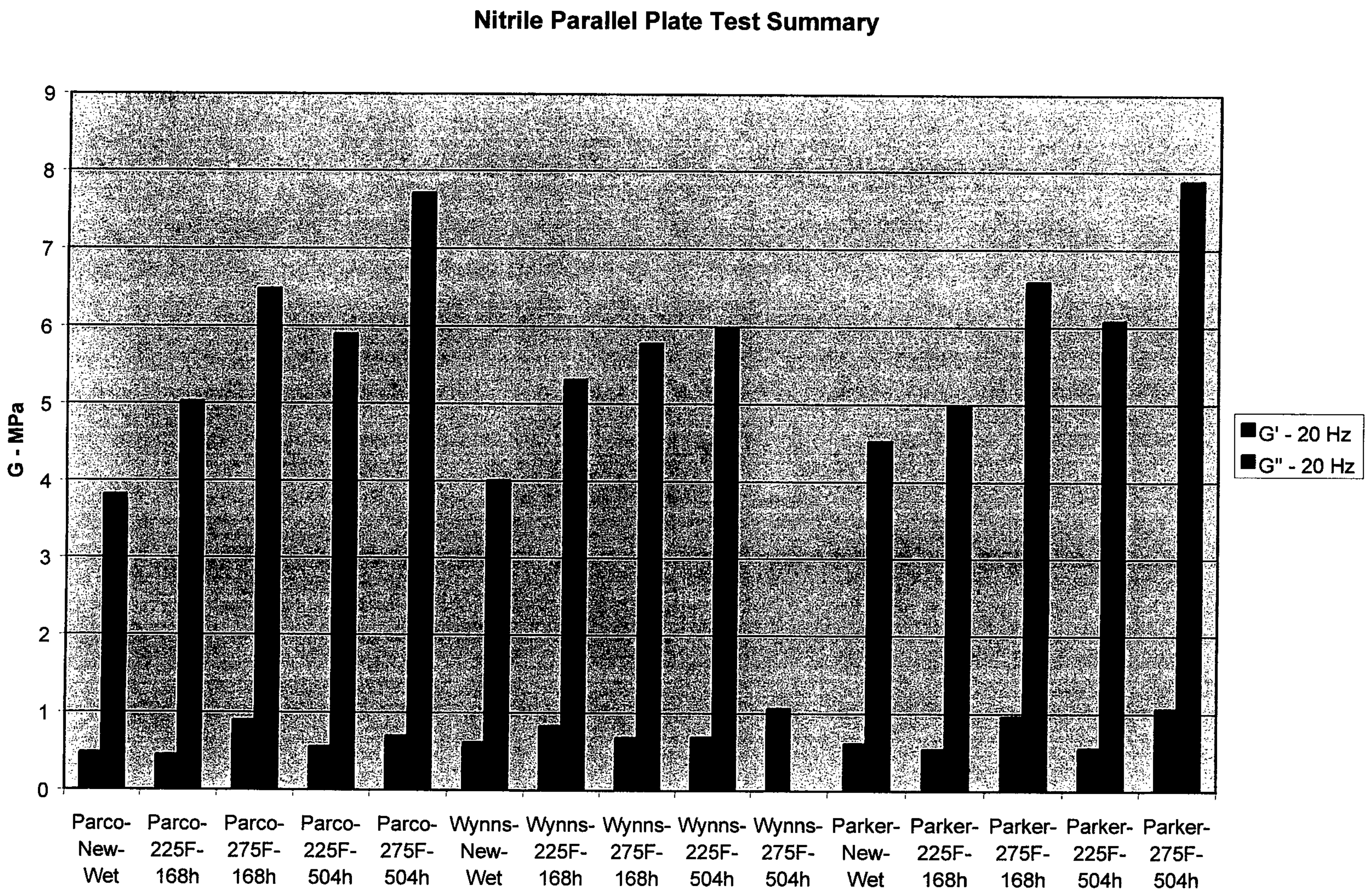


Figure 7.7: Summary of Nitrile Parallel Plate Testing, 20 Hz Data

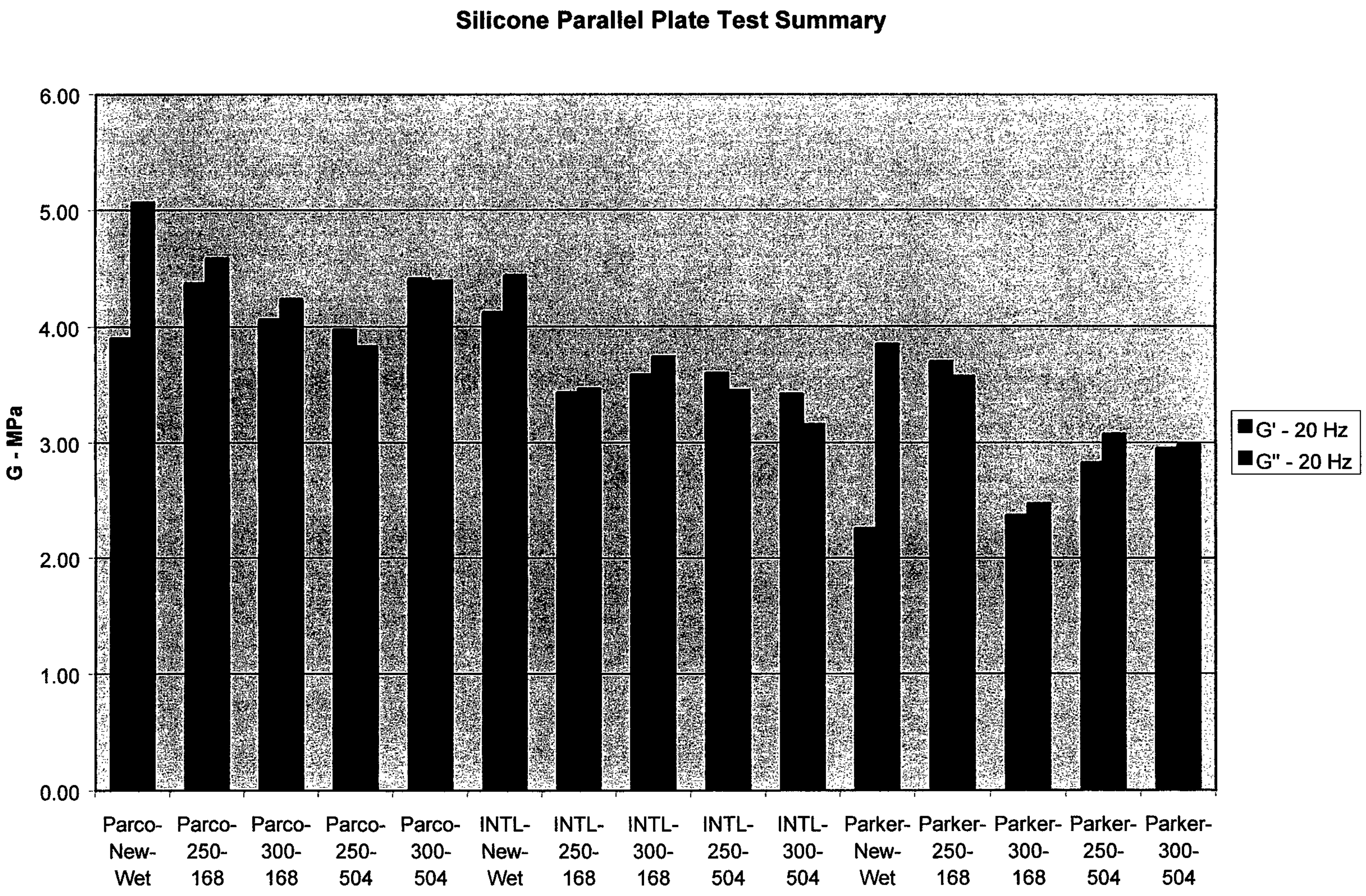


Figure 7.8: Summary of Silicone Parallel Plate Testing, 20 Hz Data

Nitrile 4000 psi 275 degF 168 hrs

Size	Material	Mfg.	Avg. Width Start 9/11-6:30p	Avg. Width End 10/02-6:30p	Avg. Height Start 9/11-6:30p	Avg. Height End 10/02-6:30p	Grams Start 9/11-6:30p	Grams End 10/02-6:30p
1-214	P83461	PARCO	0.138	0.132	0.14	0.142	1.0929	1.1191
1-214	P83461	PARCO	0.137	0.133	0.139	0.141	1.0861	1.11
1-214	P83461	PARCO	0.137	0.133	0.137	0.141	1.06	1.0868
1-214	P83461	PARCO	0.138	0.133	0.137	0.139	1.0605	1.0867
1-214	P83461	PARCO	0.137	0.133	0.139	0.141	1.0819	1.1049
214	P83461B	WYNN'S	0.137	0.134	0.139	0.14	1.0747	1.0984
214	P83461B	WYNN'S	0.137	0.134	0.139	0.14	1.0778	1.0989
214	P83461B	WYNN'S	0.137	0.132	0.139	0.141	1.0806	1.1026
2-214	NO756-75	PARKER	0.136	0.131	0.137	0.139	1.056	1.0821
2-214	NO756-75	PARKER	0.137	0.133	0.138	0.139	1.0649	1.0918
2-214	NO756-75	PARKER	0.137	0.132	0.137	0.141	1.0656	1.0905
2-214	NO756-75	PARKER	0.137	0.13	0.138	0.14	1.0625	1.0897
2-214	NO756-75	PARKER	0.137	0.133	0.138	0.14	1.0644	1.0881
214	P83461B	WYNN'S	0.137	0.134	0.139	0.139	1.0768	1.099
214	P83461B	WYNN'S	0.137	0.132	0.139	0.14	1.0831	1.0978
214	P83461B	WYNN'S	0.137	0.133	0.139	0.14	1.0818	1.1014

Figure 7.9a Example of Compression Set Test Data - Specimen Averages

Nitrile 4000 psi 275 degF 168 hrs

			Width	Height	Mass	Width	Height	Mass
			Change	Change	Change	Change	Change	Change
Size	Material	Mfg.	Width	Height	Mass	Avg.	Avg.	Avg.
1-214	P83461	PARCO	-0.0455	0.0141	0.0234			
1-214	P83461	PARCO	-0.0301	0.0142	0.0215			
1-214	P83461	PARCO	-0.0301	0.0284	0.0247			
1-214	P83461	PARCO	-0.0376	0.0144	0.0241			
1-214	P83461	PARCO	-0.0301	0.0142	0.0208	-0.035	0.017	0.186
214	P83461B	WYNN'S	-0.0224	0.0071	0.0216			
214	P83461B	WYNN'S	-0.0224	0.0071	0.0192			
214	P83461B	WYNN'S	-0.0379	0.0142	0.0200	-0.028	0.009	0.02
2-214	NO756-75	PARKER	-0.0382	0.0144	0.0241			
2-214	NO756-75	PARKER	-0.0301	0.0072	0.0246			
2-214	NO756-75	PARKER	-0.0379	0.0284	0.0228			
2-214	NO756-75	PARKER	-0.0538	0.0143	0.0250			
2-214	NO756-75	PARKER	-0.0301	0.0143	0.0218	-0.038	0.016	0.024
214	P83461B	WYNN'S	-0.0224	0	0.0202			
214	P83461B	WYNN'S	-0.0379	0.0071	0.0134			
214	P83461B	WYNN'S	-0.0301	0.0071	0.0178	-0.03	0.005	0.017
	Total	Average	-0.034	0.013	0.073	-0.033	0.012	0.062

Figure 7.9b Example of Compression Set Test Data - Material Averages

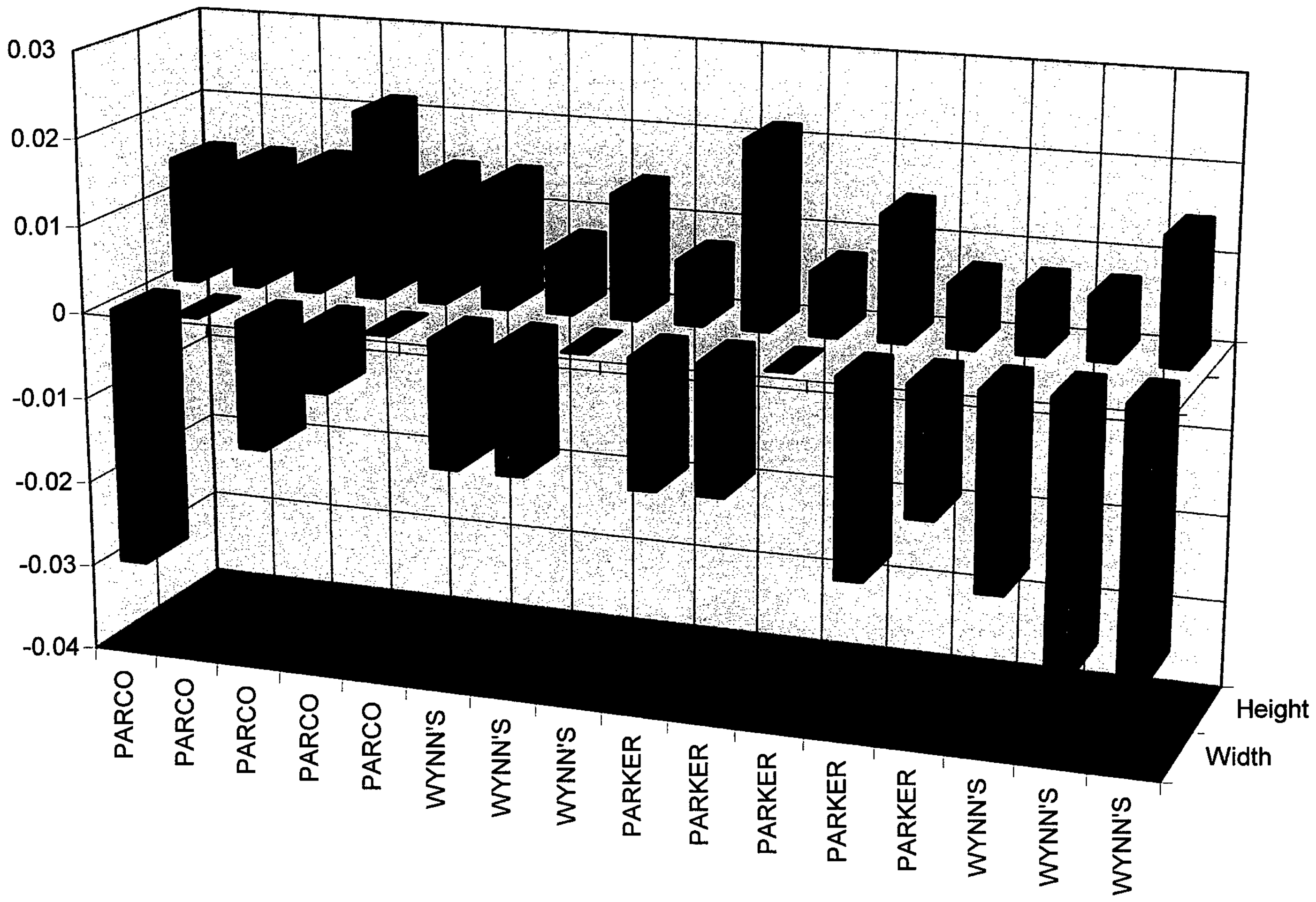


Fig. 7.10 Nitrile Compression Set Exposure for 168 hrs at 225 F and 4000 psi

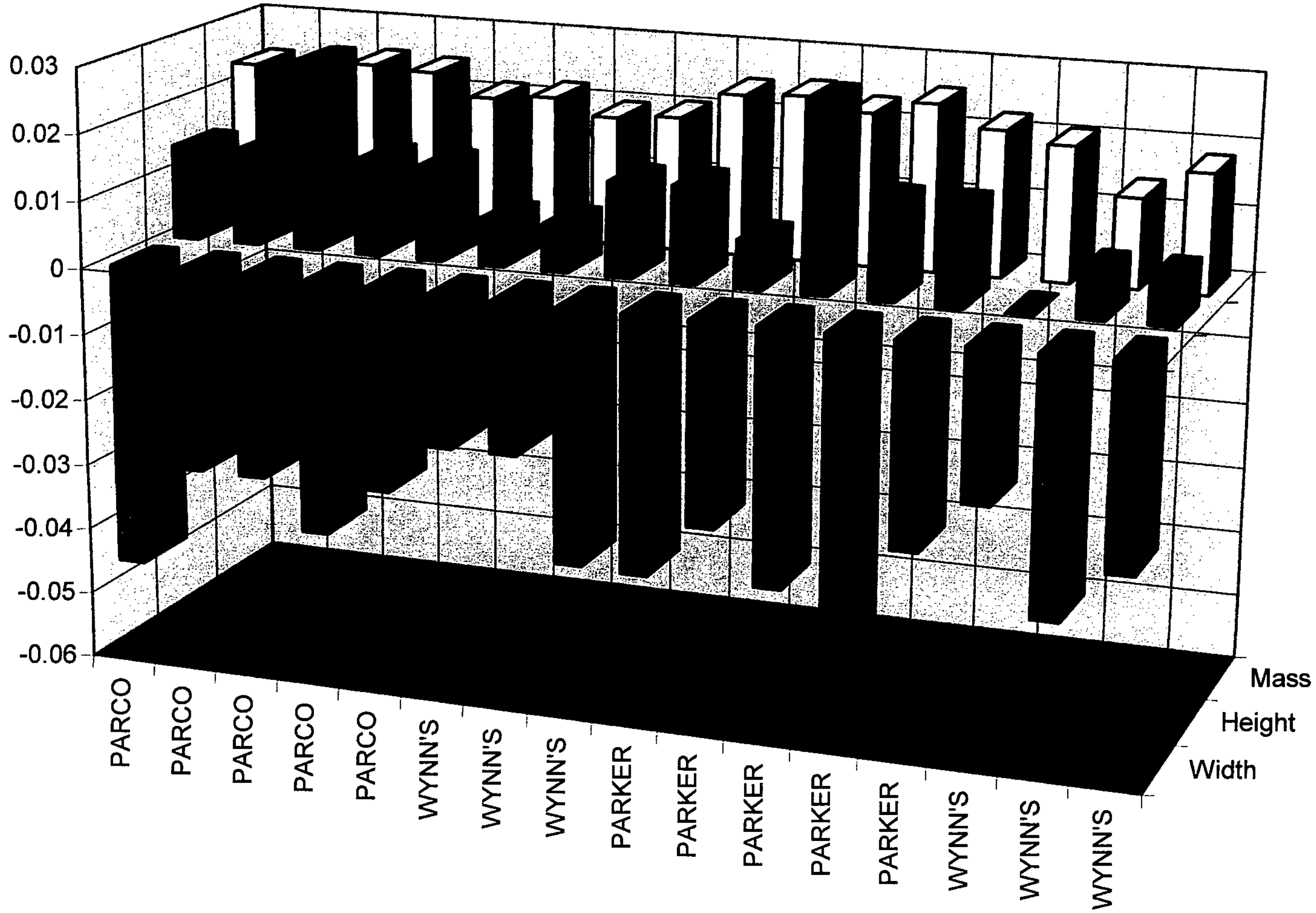


Fig. 7.11 Nitrile Compression Set Exposure for 168 hrs at 275 F and 4000psi

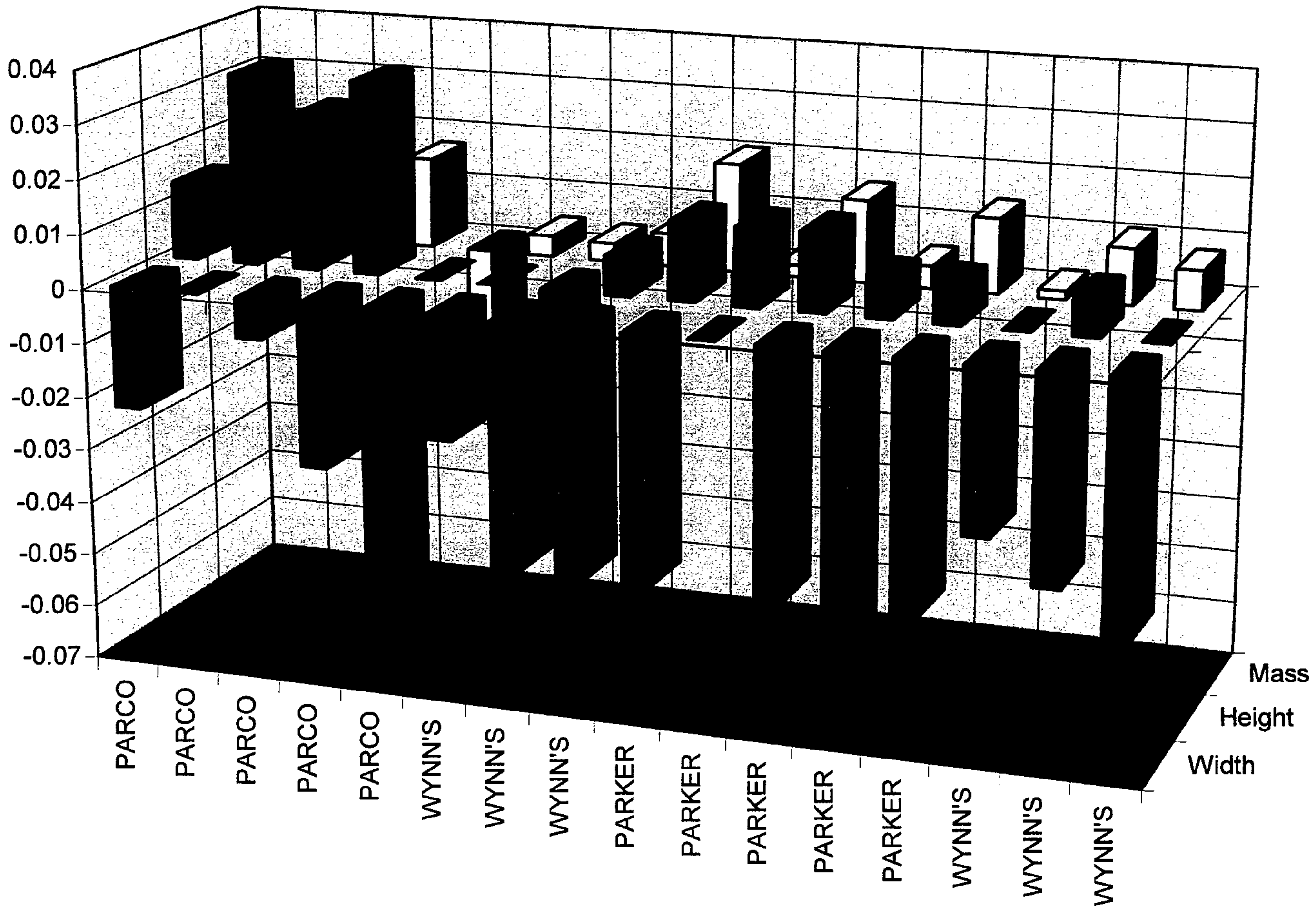


Fig. 7.12 Nitrile Compressin Set Exposure for 504 hrs at 225 F and 4000 psi

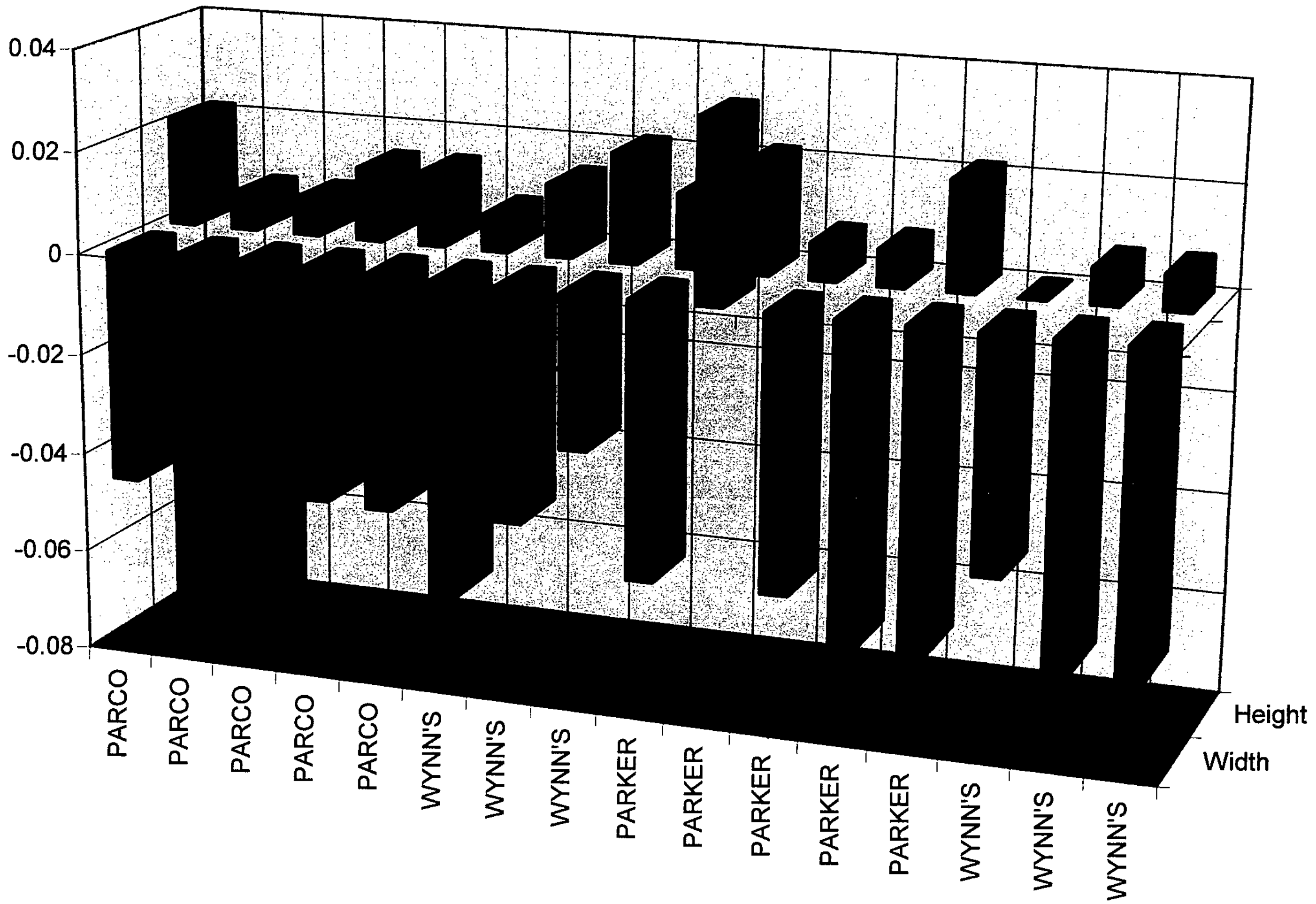


Fig. 7.13 Nitrile Compression Set Exposure for 504 hrs at 275 F and 4000 psi

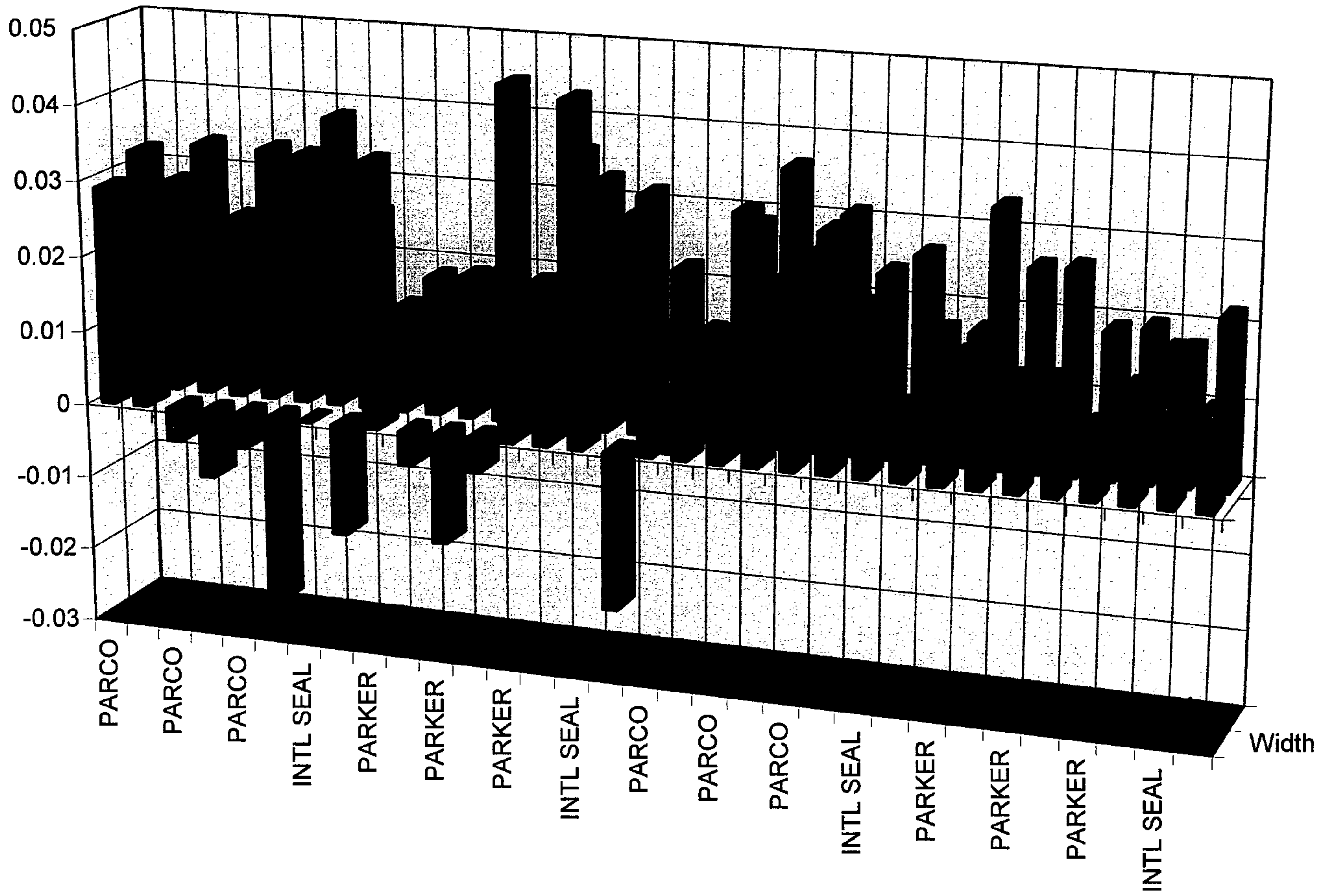


Fig. 7.14 Silicone Compression Set Exposure for 168 hrs at 250 F and 900 psi

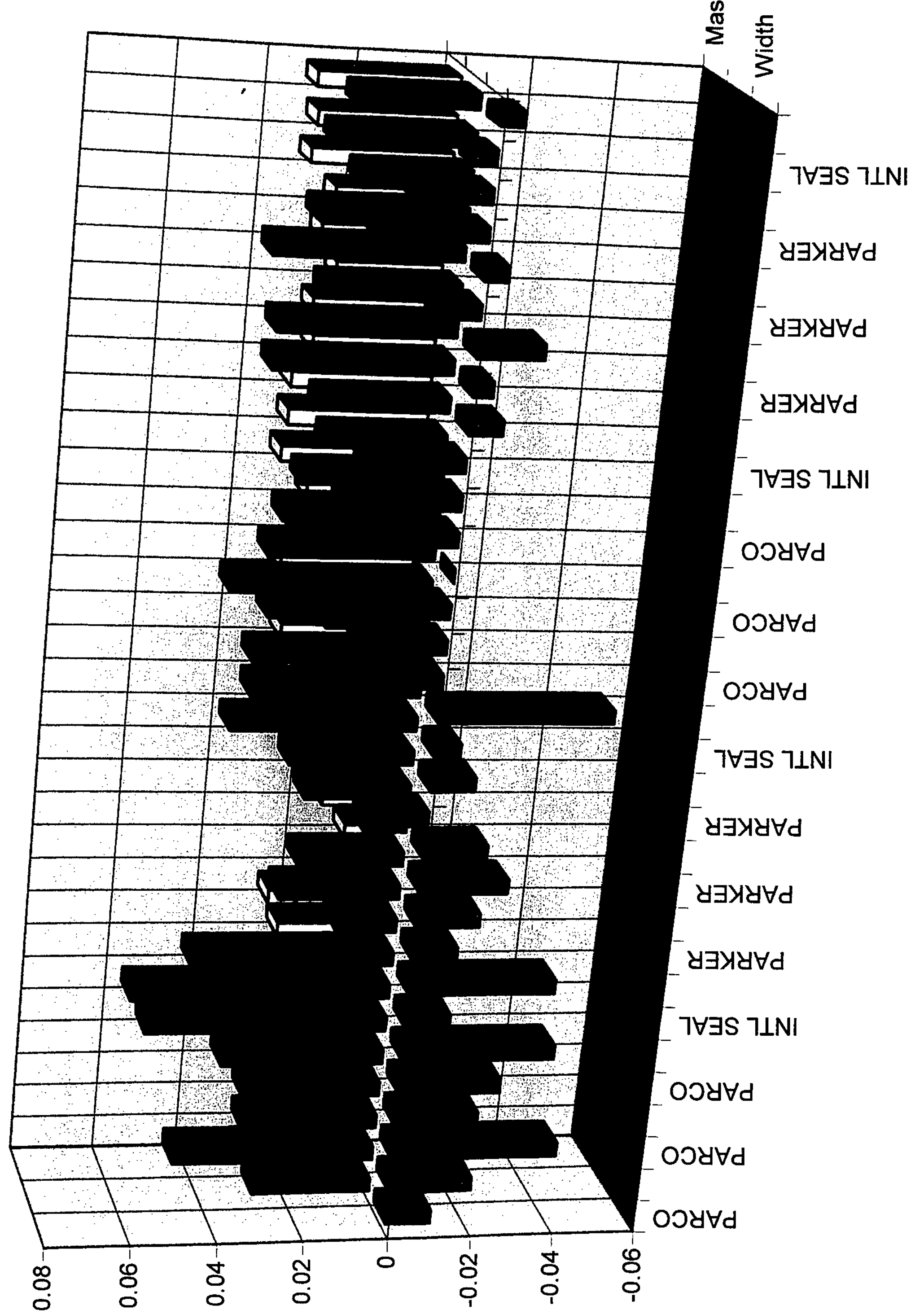


Fig 7.15 Silicone Compression Set Exposure for 168 hrs at 300 F and 900 psi

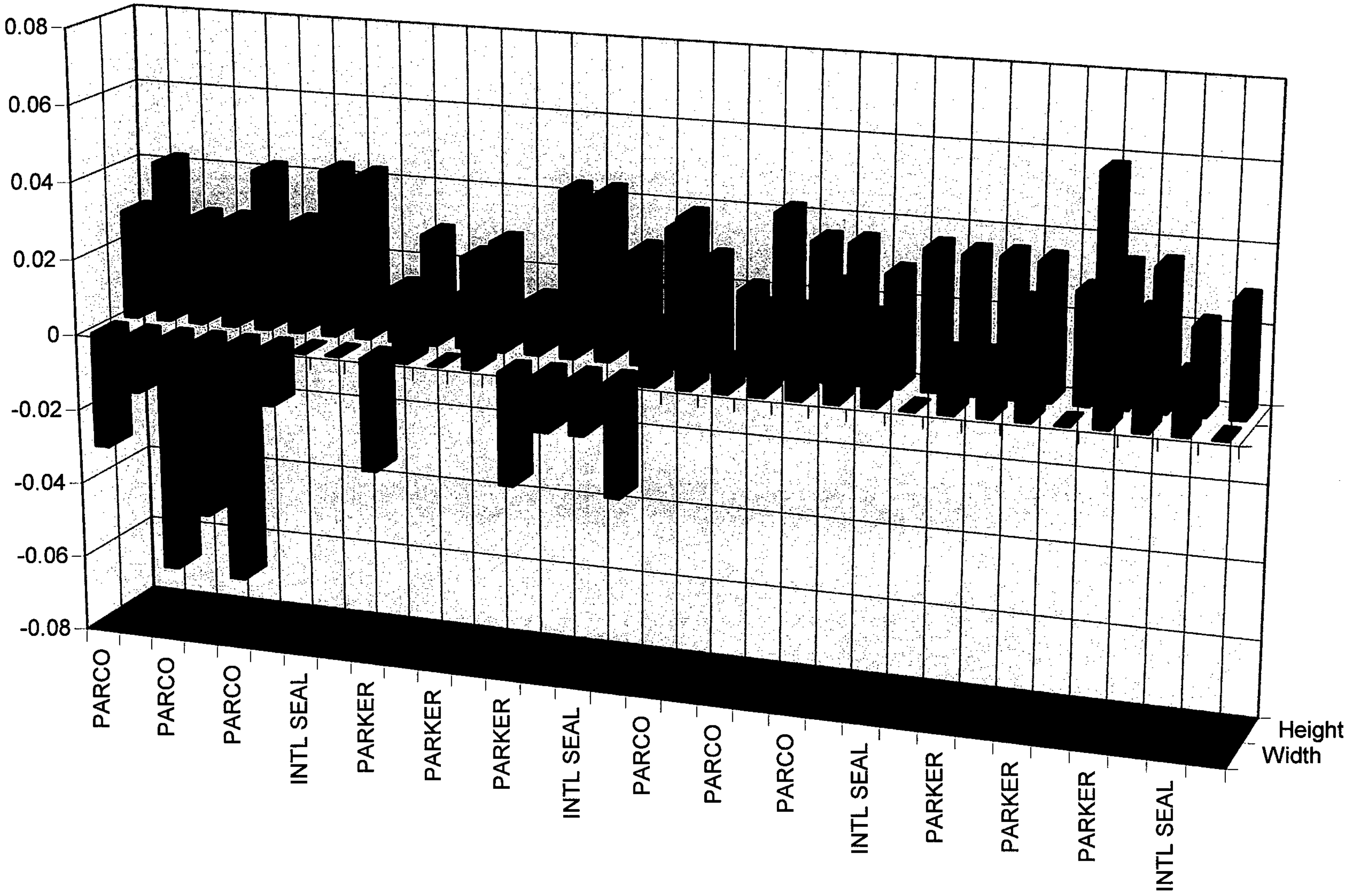


Fig. 7.16 Silicone Compression Set Exposure for 504 hrs at 250 F and 900 psi

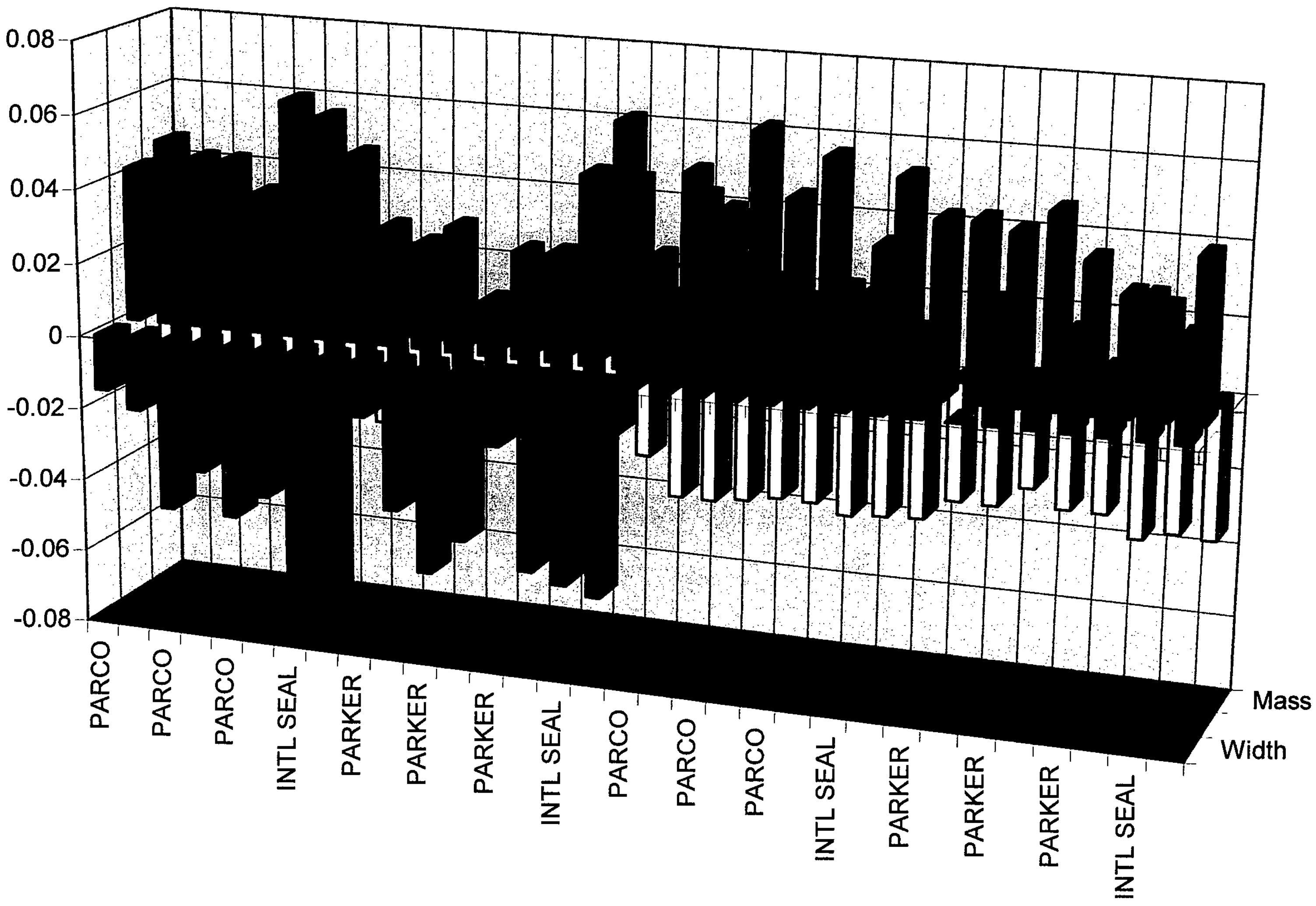


Fig 7.17 Silicone Compression Set Exposure for 504 hrs at 300 F and 900 psi

				Temp	Pressure	Time	Start	Width	Height	Start	End	Width	Height	End	ID_Code
Size	Mfg	Material	Position	(F)	(psi)	(Hr)	Date	Y avg	X avg	Weight	Date	Y avg	X avg	Weight	
										(g)				(g)	
-214	Parco	Nitrile	New, Dry			0									NA_Dry
-214	Parco	Nitrile	New, Wet			0									NA_Wet
-214	Parco	Nitrile	1	225	4000	168	29-Oct	0.1377	0.1380		5-Nov	0.1340	0.1403	1.1020	NAA411G
-214	Parco	Nitrile	2	225	4000	168	29-Oct	0.1370	0.1380		5-Nov	0.1370	0.1400	1.0940	NAA421G
-214	Parco	Nitrile	3	225	4000	168	29-Oct	0.1373	0.1370		5-Nov	0.1350	0.1393	1.0926	NAA431G
-214	Parco	Nitrile	1	275	4000	168	10-Nov	0.1383	0.1397	1.0929	17-Nov	0.1323	0.1417	1.1191	NAC411H
-214	Parco	Nitrile	2	275	4000	168	10-Nov	0.1373	0.1390	1.0861	17-Nov	0.1330	0.1413	1.1100	NAC421H
-214	Parco	Nitrile	3	275	4000	168	10-Nov	0.1373	0.1367	1.0600	17-Nov	0.1327	0.1407	1.1680	NAC431H
-214	Parco	Nitrile	1	225	4000	504	11-Sep	0.1370	0.1377	1.0789	2-Oct	0.1337	0.1400	1.1030	NAA413E
-214	Parco	Nitrile	2	225	4000	504	11-Sep	0.1380	0.1373	1.0775	2-Oct	0.1377	0.1423	1.0943	NAA423E
-214	Parco	Nitrile	3	225	4000	504	11-Sep	0.1377	0.1373	1.0706	2-Oct	0.1373	0.1413	1.0933	NAA423E
-214	Parco	Nitrile	1	275	4000	504	6-Oct	0.1367	0.1383	1.0828	27-Oct	0.1310	0.1413	1.1095	NAC413F
-214	Parco	Nitrile	2	275	4000	504	6-Oct	0.1373	0.1373	1.0694	27-Oct	0.1280	0.1377	1.0968	NAC423F
-214	Parco	Nitrile	3	275	4000	504	6-Oct	0.1373	0.1380	1.0822	27-Oct	0.1277	0.1393	1.1045	NAC433F
-214	Wynns	Nitrile	New, Dry			0									ND_Dry
-214	Wynns	Nitrile	New, Wet			0									ND_Wet
-214	Wynns	Nitrile	6	225	4000	168	29-Oct	0.1370	0.1380		5-Nov	0.1353	0.1400	1.0909	NDA4F1G
-214	Wynns	Nitrile	7	225	4000	168	29-Oct	0.1367	0.1380		5-Nov	0.1347	0.1390	1.0922	NDA4G1G
-214	Wynns	Nitrile	8	225	4000	168	29-Oct	0.1370	0.1390		5-Nov	0.1367	0.1410	1.0918	NDA4H1G
-214	Wynns	Nitrile	6	275	4000	168	10-Nov	0.1367	0.1390	1.0747	17-Nov	0.1337	0.1403	1.0984	NDC4F1H
-214	Wynns	Nitrile	7	275	4000	168	10-Nov	0.1373	0.1387	1.0778	17-Nov	0.1337	0.1067	1.0989	NDC4G1H
-214	Wynns	Nitrile	8	275	4000	168	10-Nov	0.1373	0.1390	1.0806	17-Nov	0.1323	0.1410	1.1026	NDC4H1H
-214	Wynns	Nitrile	6	225	4000	504	11-Sep	0.1357	0.1380	1.0697	2-Oct	0.1330	0.1383	1.0738	NDA463E
-214	Wynns	Nitrile	7	225	4000	504	11-Sep	0.1373	0.1387	1.0814	2-Oct	0.1313	0.0717	1.0850	NDA473E
-214	Wynns	Nitrile	8	225	4000	504	11-Sep	0.1367	0.1380	1.0731	2-Oct	0.1310	0.1057	1.0796	NDA473E
-214	Wynns	Nitrile	6	275	4000	504	6-Oct	0.1370	0.1390	1.0824	27-Oct	0.1303	0.1400	1.1019	NDC463F
-214	Wynns	Nitrile	7	275	4000	504	6-Oct	0.1370	0.1390	1.0783	27-Oct	0.1307	0.1407	1.1008	NDC473F
-214	Wynns	Nitrile	8	275	4000	504	6-Oct	0.1370	0.1380	1.0789	27-Oct	0.1330	0.1410	1.1025	NDC483F
-214	Parker	Nitrile	New, Dry			0									NC_Dry
-214	Parker	Nitrile	New, Wet			0									NC_Wet
-214	Parker	Nitrile	1P	225	4000	168	29-Oct	0.1363	0.1380		5-Nov	0.1340	0.1393	1.0892	NCA4A1G
-214	Parker	Nitrile	2P	225	4000	168	29-Oct	0.1370	0.1367		5-Nov	0.1353	0.1400	1.0823	NCA4B1G
-214	Parker	Nitrile	3P	225	4000	168	29-Oct	0.1367	0.1377		5-Nov	0.1367	0.1393	1.0828	NCA4C1G
-214	Parker	Nitrile	1P	275	4000	168	10-Nov	0.1363	0.1373	1.0560	17-Nov	0.1310	0.1393	1.0821	NCC4A1H
-214	Parker	Nitrile	2P	275	4000	168	10-Nov	0.1370	0.1380	1.0649	17-Nov	0.1327	0.1393	1.0918	NCC4B1H
-214	Parker	Nitrile	3P	275	4000	168	10-Nov	0.1370	0.1373	1.0656	17-Nov	0.1320	0.1407	1.0905	NCC4C1H
-214	Parker	Nitrile	1P	225	4000	504	11-Sep	0.1370	0.1367	1.0531	2-Oct	0.1307	0.1387	1.0746	NCA4A3E
-214	Parker	Nitrile	2P	225	4000	504	11-Sep	0.1373	0.1377	1.0670	2-Oct	0.1373	0.1403	1.0692	NCA4B3E
-214	Parker	Nitrile	3P	225	4000	504	11-Sep	0.1370	0.1373	1.0604	2-Oct	0.1310	0.1387	1.0767	NCA4C3E
-214	Parker	Nitrile	1P	275	4000	504	6-Oct	0.1360	0.1370	1.0576	27-Oct	0.1293	0.1393	1.0792	NCC4A3F
-214	Parker	Nitrile	2P	275	4000	504	6-Oct	0.1373	0.1373	1.0635	27-Oct	0.1417	0.1400	1.0889	NCC4B3F
-214	Parker	Nitrile	3P	275	4000	504	6-Oct	0.1373	0.1377	1.0668	27-Oct	0.1300	0.1387	1.0879	NCC4C3F

Table 7.1: Master Aging and Compression Set Data for Nitrile O-Rings.

			Temp	Pressure	Time		Rebound 1			Rebound 2			Rebound 3		ID_Code
Size	Mfg	Material	(F)	(psi)	(Hr)	Freq_1	Gp1	Gpp1	Freq_2	Gp2	Gpp2	Freq_3	Gp3	Gpp3	
						(Hz)	(MPa)	(MPa)	(Hz)	(MPa)	(MPa)	(Hz)	(MPa)	(MPa)	
-214	Parco	Nitrile			0	27.6	2.443	0.777	22.5	1.638	0.465	18.7	1.140	0.322	NA_Dry
-214	Parco	Nitrile			0	22.9	1.640	0.639	20.2	1.324	0.373	13.5	0.585	0.197	NA_Wet
-214	Parco	Nitrile	225	4000	168	25.4	2.034	0.745	24.0	1.864	0.512	17.0	9.256	0.304	NAA411G
-214	Parco	Nitrile	225	4000	168	27.1	0.233	0.841	25.1	2.054	0.563	19.1	1.167	0.388	NAA421G
-214	Parco	Nitrile	225	4000	168	25.8	2.106	0.764	21.7	1.534	0.412	20.2	1.309	0.415	NAA431G
-214	Parco	Nitrile	275	4000	168	30.6	2.991	1.012	23.5	1.786	0.494	19.7	1.249	0.406	NAC411H
-214	Parco	Nitrile	275	4000	168	36.0	4.117	1.461	26.0	2.196	0.587	19.5	1.209	0.410	NAC421H
-214	Parco	Nitrile	275	4000	168	28.7	2.593	0.994	26.9	2.338	0.667	23.6	1.797	0.522	NAC431H
-214	Parco	Nitrile	225	4000	504	24.0	1.844	0.618	21.9	1.555	0.445	19.9	1.266	0.424	NAA413E
-214	Parco	Nitrile	225	4000	504	2.9	2.748	0.910	27.7	2.487	0.001	22.5	1.630	0.479	NAA423E
-214	Parco	Nitrile	225	4000	504	26.3	2.187	0.807	19.1	1.178	0.350	16.6	0.883	0.290	NAA423E
-214	Parco	Nitrile	275	4000	504	30.2	2.893	1.067	32.5	3.391	1.060	25.2	2.062	0.587	NAC413F
-214	Parco	Nitrile	275	4000	504	26.1	2.136	0.825	27.9	2.493	0.791	19.4	1.206	0.380	NAC423F
-214	Parco	Nitrile	275	4000	504	29.5	2.736	1.052	25.4	2.064	0.658	15.0	0.722	0.220	NAC433F
-214	Wynns	Nitrile			0	27.2	2.368	0.768	23.9	1.849	0.531	17.7	1.011	0.313	ND_Dry
-214	Wynns	Nitrile			0	23.8	1.762	0.717	15.8	0.804	0.251	14.0	0.631	0.194	ND_Wet
-214	Wynns	Nitrile	225	4000	168	30.9	3.004	1.156	25.4	2.081	0.605	20.5	1.363	0.393	NDA4F1G
-214	Wynns	Nitrile	225	4000	168	24.9	1.961	0.696	23.5	1.798	0.507	16.8	0.898	0.303	NDA4G1G
-214	Wynns	Nitrile	225	4000	168	30.1	2.896	0.990	22.3	1.612	0.448	16.2	1.468	0.476	NDA4H1G
-214	Wynns	Nitrile	275	4000	168	27.4	2.382	0.858	26.5	2.273	0.624	16.3	1.446	0.471	NDC4F1H
-214	Wynns	Nitrile	275	4000	168	26.4	2.212	7.846	22.9	1.703	0.449	24.3	1.899	0.594	NDC4G1H
-214	Wynns	Nitrile	275	4000	168	31.0	3.074	1.032	13.5	2.998	0.810	26.1	2.189	0.667	NDC4H1H
-214	Wynns	Nitrile	225	4000	504	26.6	2.449	0.816	27.0	2.351	0.713	20.9	1.407	0.444	NDA463E
-214	Wynns	Nitrile	225	4000	504	28.1	2.561	1.024	28.1	2.561	0.701	21.6	1.489	5.012	NDA473E
-214	Wynns	Nitrile	225	4000	504	29.2	2.710	0.977	19.1	1.170	0.363	15.5	0.769	0.240	NDA473E
-214	Wynns	Nitrile	275	4000	504	27.6	2.386	0.930	20.9	1.399	0.434	16.2	0.849	0.260	NDC463F
-214	Wynns	Nitrile	275	4000	504	23.4	1.711	0.677	19.8	1.262	0.396	14.3	0.651	0.223	NDC473F
-214	Wynns	Nitrile	275	4000	504	21.4	1.415	0.598	18.6	1.115	0.343	13.6	0.587	0.207	NDC483F
-214	Parker	Nitrile			0	35.7	2.078	0.782	27.9	2.033	0.315	15.0	0.718	0.235	NC_Dry
-214	Parker	Nitrile			0	21.4	1.429	0.567	15.5	0.779	0.236	11.9	0.451	0.152	NC_Wet
-214	Parker	Nitrile	225	4000	168	34.3	3.712	1.383	22.5	1.647	0.445	23.5	1.795	0.517	NCA4A1G
-214	Parker	Nitrile	225	4000	168	24.1	1.845	0.654	23.6	1.816	0.486	19.2	1.182	0.378	NCA4B1G
-214	Parker	Nitrile	225	4000	168	28.3	2.531	0.957	29.4	2.810	0.778	22.5	1.634	0.501	NCA4C1G
-214	Parker	Nitrile	275	4000	168	28.3	2.517	0.986	24.6	1.963	0.579	20.4	1.343	0.382	NCC4A1H
-214	Parker	Nitrile	275	4000	168	29.6	2.762	1.056	27.0	2.363	0.687	18.0	1.049	0.320	NCC4B1H
-214	Parker	Nitrile	275	4000	168	28.9	2.626	0.988	28.5	2.635	0.774	20.0	1.294	0.396	NCC4C1H
-214	Parker	Nitrile	225	4000	504	27.3	2.359	0.863	22.9	1.688	0.514	15.7	7.936	0.255	NCA4A3E
-214	Parker	Nitrile	225	4000	504	26.1	2.148	0.788	20.0	1.293	0.385	17.6	1.002	0.305	NCA4B3E
-214	Parker	Nitrile	225	4000	504	28.2	2.516	0.915	24.1	1.880	0.562	17.2	0.944	0.304	NCA4C3E
-214	Parker	Nitrile	275	4000	504	22.9	1.651	0.633	22.5	1.624	0.529	12.9	0.533	0.186	NCC4A3F
-214	Parker	Nitrile	275	4000	504	27.8	2.395	1.017	20.3	1.319	0.427	16.2	0.842	0.278	NCC4B3F
-214	Parker	Nitrile	275	4000	504	27.2	2.293	0.967	27.2	2.386	0.741	19.3	1.189	0.388	NCC4C3F

Table 7.1a: Master Aging and Rebound Data for Nitrile O-Rings.

				Temp	Pressure	Time	Pplate (1 Hz)		Pplate (10 Hz)		Pplate (20 Hz)		Pplate (40 Hz)		ID_Code
Size	Mfg	Material	Position	(F)	(psi)	(Hr)	Gp_1	Gpp_1	Gp_10	Gpp_10	Gp_20	Gpp_20	Gp_40	Gpp_40	
							(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	
-214	Parco	Nitrile	New, Dry			0	4.728	0.501	5.033	0.073	5.261	0.510	5.365	0.037	NA_Dry
-214	Parco	Nitrile	New, Wet			0	3.613	0.470	3.810	0.062	4.001	0.045	4.069	0.027	NA_Wet
-214	Parco	Nitrile	1	225	4000	168	4.706	0.442	5.022	0.090	5.103	0.049	6.597	0.044	NAA411G
-214	Parco	Nitrile	2	225	4000	168									NAA421G
-214	Parco	Nitrile	3	225	4000	168									NAA431G
-214	Parco	Nitrile	1	275	4000	168	6.081	0.889	6.472	0.167	6.643	0.078	7.613	0.054	NAC411H
-214	Parco	Nitrile	2	275	4000	168									NAC421H
-214	Parco	Nitrile	3	275	4000	168									NAC431H
-214	Parco	Nitrile	1	225	4000	504	5.283	0.549	5.890	0.099	6.157	0.072	6.477	0.047	NAA413E
-214	Parco	Nitrile	2	225	4000	504									NAA423E
-214	Parco	Nitrile	3	225	4000	504									NAA423E
-214	Parco	Nitrile	1	275	4000	504	7.076	0.692	7.703	0.187	7.999	0.088	8.151	0.070	NAC413F
-214	Parco	Nitrile	2	275	4000	504									NAC423F
-214	Parco	Nitrile	3	275	4000	504									NAC433F
-214	Wynns	Nitrile	New, Dry			0	4.312	0.600	4.724	0.095	4.939	0.055	5.203	0.039	ND_Dry
-214	Wynns	Nitrile	New, Wet			0	3.876	0.610	4.002	0.094	4.113	0.049	4.310	0.028	ND_Wet
-214	Wynns	Nitrile	6	225	4000	168	4.893	0.825	5.307	0.120	5.442	0.069	5.553	0.041	NDA4F1G
-214	Wynns	Nitrile	7	225	4000	168									NDA4G1G
-214	Wynns	Nitrile	8	225	4000	168									NDA4H1G
-214	Wynns	Nitrile	6	275	4000	168	5.207	0.672	5.772	0.115	5.939	0.077	6.104	0.049	NDC4F1H
-214	Wynns	Nitrile	7	275	4000	168									NDC4G1H
-214	Wynns	Nitrile	8	275	4000	168									NDC4H1H
-214	Wynns	Nitrile	6	225	4000	504	5.493	0.682	5.972	0.117	6.264	0.073	6.385	0.051	NDA463E
-214	Wynns	Nitrile	7	225	4000	504									NDA473E
-214	Wynns	Nitrile	8	225	4000	504									NDA473E
-214	Wynns	Nitrile	6	275	4000	504	7.356	1.054	–	–	–	–	11.031	0.077	NDC463F
-214	Wynns	Nitrile	7	275	4000	504									NDC473F
-214	Wynns	Nitrile	8	275	4000	504									NDC483F
-214	Parker	Nitrile	New, Dry			0	4.782	0.777	5.341	0.103	5.579	0.060	5.658	0.041	NC_Dry
-214	Parker	Nitrile	New, Wet			0	4.312	0.605	4.522	0.094	4.689	0.059	4.783	0.036	NC_Wet
-214	Parker	Nitrile	1P	225	4000	168	6.547	0.535	4.981	0.097	6.811	0.074	5.407	0.020	NCA4A1G
-214	Parker	Nitrile	2P	225	4000	168									NCA4B1G
-214	Parker	Nitrile	3P	225	4000	168									NCA4C1G
-214	Parker	Nitrile	1P	275	4000	168	6.062	0.954	6.574	0.140	6.685	0.080	6.986	0.056	NCC4A1H
-214	Parker	Nitrile	2P	275	4000	168									NCC4B1H
-214	Parker	Nitrile	3P	275	4000	168									NCC4C1H
-214	Parker	Nitrile	1P	225	4000	504	5.588	0.547	6.074	0.093	6.377	0.075	6.551	0.051	NCA4A3E
-214	Parker	Nitrile	2P	225	4000	504									NCA4B3E
-214	Parker	Nitrile	3P	225	4000	504									NCA4C3E
-214	Parker	Nitrile	1P	275	4000	504	7.097	1.053	7.875	0.144	8.261	0.099	8.599	0.057	NCC4A3F
-214	Parker	Nitrile	2P	275	4000	504									NCC4B3F
-214	Parker	Nitrile	3P	275	4000	504									NCC4C3F

Table 7.1b: Master Aging and Parallel Plate Data for Nitrile O-Rings.

				Temp	Pressure	Time	Start	Width	Height	Start	End	Width	Height	End	ID_Code
Size	Mfg	Material	Position	(F)	(psi)	(Hr)	Date	Y avg	X avg	Weight	Date	Y avg	X avg	Weight	
								(in)	(in)	(g)		(in)	(in)	(g)	
-214	Parco	Silicone	New, Dry			0									SA_Dry
-214	Parco	Silicone	New, Wet			0									SA_Wet
-214	Parco	Silicone	1	250	900	168	18-Jul	0.1350	0.1380	1.3458	23-Jul	0.1397	0.1377	1.3830	SAB911B
-214	Parco	Silicone	2	250	900	168	18-Jul	0.1353	0.1380	1.3553	23-Jul	0.1387	0.1400	1.3931	SAB921B
-214	Parco	Silicone	3	250	900	168	18-Jul	0.1357	0.1380	1.3580	23-Jul	0.1380	0.1423	1.3961	SAB931B
-214	Parco	Silicone	1	300	900	168	21-Aug	0.1013	0.1377	1.3525	28-Aug	0.1380	0.1433	1.3929	SAD911D
-214	Parco	Silicone	2	300	900	168	21-Aug	0.1350	0.1387	1.3605	28-Aug	0.1377	0.1420	1.4043	SAD921D
-214	Parco	Silicone	3	300	900	168	21-Aug	0.1340	0.1343	1.2955	28-Aug	0.1397	0.1407	1.3363	SAD931D
-214	Parco	Silicone	1	250	900	504	18-Jun	0.1343	0.1363		10-Jul	0.1357	0.1410		SAB913A
-214	Parco	Silicone	2	250	900	504	18-Jun	0.1343	0.1370		10-Jul	0.1400	0.1410		SAB923A
-214	Parco	Silicone	3	250	900	504	18-Jun	0.1347	0.1390		10-Jul	0.1363	0.1420		SAB933A
-214	Parco	Silicone	1	300	900	504	29-Jul	0.1353	0.1377	1.3522	20-Aug	0.1430	0.1420	1.4050	SAD913C
-214	Parco	Silicone	2	300	900	504	29-Jul	0.1353	0.1370	1.3333	20-Aug	0.1387	0.1447	1.3855	SAD923C
-214	Parco	Silicone	3	300	900	504	29-Jul	0.1347	0.1370	1.3392	20-Aug	0.1420	0.1433	1.3901	SAD933C
-214	IntlSeal	Silicone	New, Dry			0									SB_Dry
-214	IntlSeal	Silicone	New, Wet			0									SB_Wet
-214	IntlSeal	Silicone	6	250	900	168	18-Jul	0.1347	0.1347	1.2906	23-Jul	0.1390	0.1390	1.3312	SBB961B
-214	IntlSeal	Silicone	7	250	900	168	18-Jul	0.1340	0.1350	1.2918	23-Jul	0.1370	0.1383	1.3319	SBB971B
-214	IntlSeal	Silicone	8	250	900	168	18-Jul	0.1340	0.1353	1.2838	23-Jul	0.1353	0.1390	1.3230	SBB981B
-214	IntlSeal	Silicone	6	300	900	168	21-Aug	0.1337	0.1343	1.2756	28-Aug	0.1373	0.1387	1.3187	SBD9F1D
-214	IntlSeal	Silicone	7	300	900	168	21-Aug	0.1330	0.1343	1.2884	28-Aug	0.1360	0.1380	1.3307	SBD9G1D
-214	IntlSeal	Silicone	8	300	900	168	21-Aug	0.1340	0.1333	1.2738	28-Aug	0.1330	0.1373	1.3156	SBD9H1D
-214	IntlSeal	Silicone	6	250	900	504	18-Jun	0.1347	0.1357		10-Jul	0.1393	0.1407		SBB963A
-214	IntlSeal	Silicone	7	250	900	504	18-Jun	0.1353	0.1363		10-Jul	0.1377	0.1397		SBB973A
-214	IntlSeal	Silicone	8	250	900	504	18-Jun	0.1337	0.1350		10-Jul	0.1337	0.1400		SBB983A
-214	IntlSeal	Silicone	6	300	900	504	29-Jul	0.1333	0.1317	1.2832	20-Aug	0.1370	0.1433	1.3334	SBD9F3C
-214	IntlSeal	Silicone	7	300	900	504	29-Jul	0.1337	0.1333	1.2574	20-Aug	0.1380	0.1367	1.3057	SBD9G3C
-214	IntlSeal	Silicone	8	300	900	504	29-Jul	0.1337	0.1337	1.2697	20-Aug	0.1397	0.1417	1.3178	SBD9H3C
-214	Parker	Silicone	New, Dry			0									SC_Dry
-214	Parker	Silicone	New, Wet			0									SC_Wet
-214	Parker	Silicone	1P	250	900	168	18-Jul	0.1350	0.1383	1.3266	23-Jul	0.1377	0.1403	1.3573	SCB9A1B
-214	Parker	Silicone	2P	250	900	168	18-Jul	0.1330	0.1363	1.2947	23-Jul	0.1357	0.1410	1.3274	SCB9B1B
-214	Parker	Silicone	3P	250	900	168	18-Jul	0.1363	0.1383	1.3443	23-Jul	0.1383	0.1420	1.3755	SCB9C1B
-214	Parker	Silicone	1P	300	900	168	21-Aug	0.1343	0.1353	1.2917	28-Aug	0.1337	0.1410	1.3284	SCD9A1D
-214	Parker	Silicone	2P	300	900	168	21-Aug	0.1330	0.1363	1.2920	28-Aug	0.1310	0.1420	1.3297	SCD9B1D
-214	Parker	Silicone	3P	300	900	168	21-Aug	0.1347	0.1380	1.3402	28-Aug	0.1360	0.1423	1.3728	SCD9C1D
-214	Parker	Silicone	1P	250	900	504	18-Jun	0.1353	0.1383		10-Jul	0.1370	0.1430		SCB9A3A
-214	Parker	Silicone	2P	250	900	504	18-Jun	0.1323	0.1367		10-Jul	0.1340	0.1417		SCB9B3A
-214	Parker	Silicone	3P	250	900	504	18-Jun	0.1357	0.1387		10-Jul	0.1400	0.1440		SCB9C3A
-214	Parker	Silicone	1P	300	900	504	29-Jul	0.1340	0.1373	1.3308	20-Aug	0.1370	0.1440	1.3735	SCD9A3C
-214	Parker	Silicone	2P	300	900	504	29-Jul	0.1327	0.1360	1.2830	20-Aug	0.1327	0.1427	1.3244	SCD9B3C
-214	Parker	Silicone	3P	300	900	504	29-Jul	0.1347	0.1363	1.3279	20-Aug	0.1390	0.1427	1.3631	SCD9C3C

Table 7.2: Master Aging and Compression Set Data for Silicone O-Rings.

				Temp	Pressure	Time	Rebound 1			Rebound 2			Rebound 3			ID_Code
Size	Mfg	Material	Position	(F)	(psi)	(Hr)	Freq_1	Gp1	Gpp1	Freq_2	Gp2	Gpp2	Freq_3	Gp3	Gpp3	
							(Hz)	(MPa)	(MPa)	(Hz)	(MPa)	(MPa)	(Hz)	(MPa)	(MPa)	
-214	Parco	Silicone	New, Dry			0	22.8	1.524	0.876	13.7	0.552	0.300	26.3	2.015	1.156	SA_Dry
-214	Parco	Silicone	New, Wet			0	28.9	2.461	1.355	22.3	1.520	0.702	15.8	0.761	0.367	SA_Wet
-214	Parco	Silicone	1	250	900	168	31.6	2.879	1.751	22.8	1.542	0.884	15.8	0.746	0.391	SAB911B
-214	Parco	Silicone	2	250	900	168	29.3	2.469	1.497	23.9	1.674	0.964	15.4	0.716	0.362	SAB921B
-214	Parco	Silicone	3	250	900	168	26.4	2.027	1.172	20.2	1.210	0.655	15.3	0.691	0.381	SAB931B
-214	Parco	Silicone	1	300	900	168	31.9	2.941	1.746	26.8	2.122	1.158	23.0	1.613	0.745	SAD911D
-214	Parco	Silicone	2	300	900	168	25.5	1.876	1.141	18.8	1.030	0.600	10.6	0.335	0.176	SAD921D
-214	Parco	Silicone	3	300	900	168	29.0	2.423	1.483	24.5	1.758	1.001	15.8	0.754	0.372	SAD931D
-214	Parco	Silicone	1	250	900	504	28.6	2.359	1.418	23.7	1.639	0.940	14.5	0.641	0.300	SAB913A
-214	Parco	Silicone	2	250	900	504	29.1	2.465	1.419	22.3	1.478	0.800	14.5	0.622	0.330	SAB923A
-214	Parco	Silicone	3	250	900	504	26.4	2.025	1.198	23.1	1.581	0.850	14.5	0.639	0.315	SAB933A
-214	Parco	Silicone	1	300	900	504	26.3	2.018	1.168	22.3	1.471	0.813	12.6	0.470	0.259	SAD913C
-214	Parco	Silicone	2	300	900	504	26.9	2.114	1.200	23.6	1.638	0.905	14.4	0.618	0.330	SAD923C
-214	Parco	Silicone	3	300	900	504	27.5	2.226	1.249	18.3	0.988	0.532	14.1	0.593	0.314	SAD933C
-214	IntlSeal	Silicone	New, Dry			0	23.9	1.695	0.925	14.4	0.637	0.297	9.9	0.300	0.132	SB_Dry
-214	IntlSeal	Silicone	New, Wet			0	25.1	1.091	0.945	19.4	1.161	0.505	12.8	0.521	0.181	SB_Wet
-214	IntlSeal	Silicone	6	250	900	168	28.8	2.464	1.329	20.5	1.265	0.637	15.9	0.759	0.394	SBB961B
-214	IntlSeal	Silicone	7	250	900	168	24.6	1.759	1.041	21.0	1.357	0.615	13.7	0.555	0.295	SBB971B
-214	IntlSeal	Silicone	8	250	900	168	28.7	2.437	0.000	22.0	1.461	0.735	13.3	0.521	0.283	SBB981B
-214	IntlSeal	Silicone	6	300	900	168	25.3	1.902	1.021	24.8	1.861	0.919	18.7	1.052	0.541	SBD9F1D
-214	IntlSeal	Silicone	7	300	900	168	24.0	1.723	0.910	23.5	1.666	0.840	14.3	0.612	0.318	SBD9G1D
-214	IntlSeal	Silicone	8	300	900	168	29.9	2.641	1.452	25.7	2.007	0.961	16.4	0.806	0.423	SBD9H1D
-214	IntlSeal	Silicone	6	250	900	504	29.0	2.510	1.336	20.4	1.256	0.631	16.8	0.839	0.446	SBB963A
-214	IntlSeal	Silicone	7	250	900	504	23.7	1.672	0.900	18.2	1.001	0.497	13.6	0.548	0.287	SBB973A
-214	IntlSeal	Silicone	8	250	900	504	30.1	2.685	1.461	21.7	1.420	0.703	17.1	0.870	0.471	SBB983A
-214	IntlSeal	Silicone	6	300	900	504	26.9	2.144	1.156	20.5	1.272	0.612	18.9	1.075	0.542	SBD9F3C
-214	IntlSeal	Silicone	7	300	900	504	26.9	2.156	1.147	22.5	1.537	0.749	15.8	0.758	0.371	SBD9G3C
-214	IntlSeal	Silicone	8	300	900	504	32.0	2.965	1.755	20.6	1.305	0.572	17.7	0.952	0.462	SBD9H3C
-214	Parker	Silicone	New, Dry			0	24.2	1.601	1.142	15.6	0.701	0.434	11.7	0.411	0.208	SC_Dry
-214	Parker	Silicone	New, Wet			0	28.9	2.337	1.581	10.8	1.266	0.709	17.3	0.910	0.426	SC_Wet
-214	Parker	Silicone	1P	250	900	168	25.3	1.834	1.129	21.0	1.277	0.755	11.1	0.368	0.198	SCB9A1B
-214	Parker	Silicone	2P	250	900	168	26.3	1.871	1.385	20.1	1.124	0.766	11.3	0.000	0.000	SCB9B1B
-214	Parker	Silicone	3P	250	900	168	27.2	2.141	1.287	18.2	0.944	0.590	12.1	0.426	0.249	SCB9C1B
-214	Parker	Silicone	1P	300	900	168	28.1	2.209	1.480	23.3	1.536	0.994	14.8	0.649	0.356	SCD9A1D
-214	Parker	Silicone	2P	300	900	168	27.5	2.060	1.502	18.5	0.960	0.642	14.4	0.616	0.326	SCD9B1D
-214	Parker	Silicone	3P	300	900	168	30.7	2.712	1.634	18.5	0.987	0.591	17.4	0.905	0.479	SCD9C1D
-214	Parker	Silicone	1P	250	900	504	28.5	2.306	1.466	24.2	1.686	1.023	18.1	0.972	0.528	SCB9A3A
-214	Parker	Silicone	2P	250	900	504	31.8	2.837	1.900	26.1	1.956	1.215	16.8	0.834	0.445	SCB9B3A
-214	Parker	Silicone	3P	250	900	504	27.1	2.131	1.274	17.7	0.907	0.550	13.2	0.508	0.288	SCB9C3A
-214	Parker	Silicone	1P	300	900	504	25.0	1.753	1.159	17.0	0.813	0.532	8.4	0.207	0.121	SCD9A3C
-214	Parker	Silicone	2P	300	900	504	30.0	2.508	1.696	24.1	1.663	1.032	17.6	0.922	0.487	SCD9B3C
-214	Parker	Silicone	3P	300	900	504	26.1	1.973	1.178	18.9	1.067	0.559	11.6	0.394	0.220	SCD9C3C

Table 7.2a: Master Aging and Rebound Data for Silicone O-Rings.

				Temp	Pressure	Time	Pplate (1 Hz)		Pplate (10 Hz)		Pplate (20 Hz)		Pplate (40 Hz)		ID_Code
Size	Mfg	Material	Position	(F)	(psi)	(Hr)	Gp_1	Gpp_1	Gp_10	Gpp_10	Gp_20	Gpp_20	Gp_40	Gpp_40	
							(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	(MPa)	
-214	Parco	Silicone	New, Dry			0	3.45	3.38	3.54	3.08	3.49	3.54	3.44	3.95	SA_Dry
-214	Parco	Silicone	New, Wet			0	4.18	3.80	4.54	3.96	3.90	5.06	3.77	5.29	SA_Wet
-214	Parco	Silicone	1	250	900	168	4.54	3.65	4.42	4.80	4.57	4.69	4.52	4.76	SAB911B
-214	Parco	Silicone	2	250	900	168	4.66	3.07	4.20	4.46	4.10	4.70	4.09	5.00	SAB921B
-214	Parco	Silicone	3	250	900	168	4.47	3.76	4.38	4.30	4.45	4.36	4.00	5.04	SAB931B
-214	Parco	Silicone	1	300	900	168	4.29	3.07	4.00	3.92	4.06	4.24	3.90	4.76	SAD911D
-214	Parco	Silicone	2	300	900	168									SAD921D
-214	Parco	Silicone	3	300	900	168									SAD931D
-214	Parco	Silicone	1	250	900	504									SAB913A
-214	Parco	Silicone	2	250	900	504	4.00	2.85	3.64	3.67	4.06	3.47	4.17	3.67	SAB923A
-214	Parco	Silicone	3	250	900	504	3.79	3.31	4.18	3.79	3.89	4.18	3.90	4.62	SAB933A
-214	Parco	Silicone	1	300	900	504	4.51	3.49	4.35	4.27	4.41	4.39	4.04	5.08	SAD913C
-214	Parco	Silicone	2	300	900	504									SAD923C
-214	Parco	Silicone	3	300	900	504									SAD933C
-214	IntlSeal	Silicone	New, Dry			0	3.44	3.11	3.87	3.38	3.62	4.02	3.62	4.42	SB_Dry
-214	IntlSeal	Silicone	New, Wet			0	3.95	3.73	4.37	3.96	4.12	4.44	4.09	4.85	SB_Wet
-214	IntlSeal	Silicone	6	250	900	168	3.03	2.43	3.36	2.81	3.21	3.00	3.44	3.12	SBB961B
-214	IntlSeal	Silicone	7	250	900	168	3.40	2.96	3.69	3.22	3.57	3.72	3.20	4.29	SBB971B
-214	IntlSeal	Silicone	8	250	900	168	3.62	2.92	3.69	3.22	3.51	3.66	3.25	4.09	SBB981B
-214	IntlSeal	Silicone	6	300	900	168	3.22	2.91	3.51	3.45	3.58	3.73	3.19	4.02	SBD9F1D
-214	IntlSeal	Silicone	7	300	900	168									SBD9G1D
-214	IntlSeal	Silicone	8	300	900	168									SBD9H1D
-214	IntlSeal	Silicone	6	250	900	504	3.47	2.79	3.32	3.26	3.56	3.18	3.31	3.67	SBB963A
-214	IntlSeal	Silicone	7	250	900	504	3.54	2.96	3.35	3.38	3.60	3.38	3.58	3.55	SBB973A
-214	IntlSeal	Silicone	8	250	900	504	2.62	3.82	3.69	3.22	3.62	3.78	3.31	4.44	SBB983A
-214	IntlSeal	Silicone	6	300	900	504	3.20	3.14	3.13	3.32	3.42	3.16	3.18	3.53	SBD9F3C
-214	IntlSeal	Silicone	7	300	900	504									SBD9G3C
-214	IntlSeal	Silicone	8	300	900	504									SBD9H3C
-214	Parker	Silicone	New, Dry			0									SC_Dry
-214	Parker	Silicone	New, Wet			0	2.83	2.78	3.05	3.23	2.25	3.84	2.67	3.58	SC_Wet
-214	Parker	Silicone	1P	250	900	168	3.29	3.20	3.64	3.41	3.69	3.57	3.53	3.96	SCB9A1B
-214	Parker	Silicone	2P	250	900	168									SCB9B1B
-214	Parker	Silicone	3P	250	900	168	2.87	3.42	3.03	3.61	3.56	3.29	3.22	3.94	SCB9C1B
-214	Parker	Silicone	1P	300	900	168	1.95	2.07	2.25	2.38	2.36	2.47	1.34	1.69	SCD9A1D
-214	Parker	Silicone	2P	300	900	168									SCD9B1D
-214	Parker	Silicone	3P	300	900	168									SCD9C1D
-214	Parker	Silicone	1P	250	900	504	2.48	2.32	3.40	3.43	2.82	3.07	2.81	3.56	SCB9A3A
-214	Parker	Silicone	2P	250	900	504									SCB9B3A
-214	Parker	Silicone	3P	250	900	504									SCB9C3A
-214	Parker	Silicone	1P	300	900	504	2.50	2.45	2.94	2.77	2.95	2.98	3.00	3.33	SCD9A3C
-214	Parker	Silicone	2P	300	900	504									SCD9B3C
-214	Parker	Silicone	3P	300	900	504									SCD9C3C

Table 7.2b: Master Aging and Parallel Plate Data for Silicone O-Rings.

Seal	Aging	Applied	Width	Height	Mass
Material	Time	Temperature	Change	Change	
Change					
	Hr	F	percent	percent	percent
Nitrile	168	225	-1.4	1.3	
Nitrile	168	275	-3.4	1.3	7.3
Nitrile	504	225	-3.4	1.1	0.9
Nitrile	504	275	-4.9	1.2	
Silicone	168	250	2.1	2.2	
Silicone	168	300	3.7	3.6	2.9
Silicone	504	250	2.0	3.2	
Silicone	504	300	2.9	3.8	-3.5

Table 7.3 Compression Set Data Averaged Over Suppliers.

8.0 DISCUSSION OF RESULTS

The original intent of the research program was to develop a simple benchtop test method capable of distinguishing changes in aged O-ring seals which might serve to project relative life between manufacturers. Fluorosilicone and nitrile O-ring seals were each obtained from 3 manufacturers and artificially aged. Three different methods were explored to measure changes in the seals which might serve to predict seal performance and life,

1. Compression set,
2. Dynamic rebound,
3. Cyclic compressive force-displacement response.

Changes in seal dimensions and weight were measured as a function of the aging processes to determine whether compression set might prove a useful prediction method. Previous work and these results confirmed that compression set was incapable of distinguishing between good and poor seal materials.

Using the remaining two test methods, fundamental principles of operation were established. Preliminary hardware, test methods and analytical tools were designed and developed. However, with the current level of development, limited by the scope and funds available for the program, test methods capable of determining subtle changes of seal properties were not successfully developed.

The dynamic rebound test has potential for measuring basic rubber properties between multiple samples. However, the level of sophistication required to differentiate the expected changes of properties is beyond the capability of both the current method and the test hardware. This method could be developed into a benchtop test method that the military requires. Development of this method to provide required capabilities would require a program more focused than the current effort. It is recommended and suggested that a focused activity using a limited number of well characterized elastomeric seal samples could concentrate on refining both the hardware and the analytical software.

The cyclic compressive force-displacement method also has potential similar to the rebound method; however, development of this method into a suitable benchtop test is not expected. The cost of servohydraulic test equipment, such as the MTS, Instron or similar equipment, is incompatible with the goal of developing a simple benchtop method for evaluating seals.

9.0 SUMMARY OF LIFE PREDICTION METHOD

Two methods were examined for estimating the life of O-ring seals. The first is based on the use of PC's and the second on workstations. The PC version will be coupled to the pendulum test data, while the workstation version will be coupled with the cyclic parallel plate tests. There is no reason they cannot be reversed, but the pendulum test is particularly simple and provides a good match to the PC version. A summary of the life prediction method for the PC and workstation codes is given below.

9.1 Pendulum Test and PC Code

Three steps are required for estimating the life of an O-ring using the PC software. The pendulum test data must be gathered and interpreted. A model must be generated, and then an estimate for the long term response developed. The first two steps are readily completed, but the last will depend on a comparison across several design choices to determine which design will have the longest life.

9.1.1 Interpretation of Pendulum Data

The data from the pendulum test will consist of *tanδ* or G' and G''. The most important of these is the loss modulus, G'', as it represents the permanent deformation that could occur. In the PC code there are two parameters: 1) the relaxation factor which can be set to *tanδ*. And 2) the Young's modulus which can be taken as 3 times the storage modulus, G' as in equation (3). The relaxation time is a third constant, but we only have the storage and loss moduli. If the data for the first bounce is used for G' and G'' then we set the relaxation time to the period for the first bounce, or we must look through the remaining data for an estimate of the frequency where G'' is a maximum, and set the time equal to the period while using the values for G' and G'' at that frequency. If there is no discernable maximum then set the relaxation time to the period for the first bounce frequency, or, if possible, use a bounce with a frequency near the loading frequencies. This step will ensure that the correct response will result for that frequency.

9.1.2 PC Code Modeling

Once the properties have been determined, all that remains is obtaining the data for the O-ring and gland geometry and the loading. The geometric inputs are readily determined from the proposed design. Variations in the design geometry can be readily entered and evaluated. The loading will consist of a mean squeeze and mean pressure, and variations about the mean. The frequency of the variations must also be determined. The variations in the footprints and the wall forces should be examined to determine if the seal is likely to leak. Higher normal wall forces are desirable. High frictional forces will increase the erosion rate. Smaller footprints are more susceptible to leaking due to surface roughness.

9.1.3 Long Term Response PC Prediction

If the long term response is desired, then the appropriate relaxation time must be used, and only a long time steady state case needs to be run. The long term relaxation time is difficult to determine from typical compression set data as discussed in Section 7.5; hence, it may only be possible to determine the steady wall forces, and footprints. For accurate steady state estimates, the long time G' and G'' must be known, or equivalently a long time relaxation test must be performed.

9.2 Pendulum Test and PC Code

Again, three steps are required for estimating the life of an O-ring using the workstation software. The cyclic parallel plate data must be gathered and interpreted. A finite element model must be generated, and then an estimate for the long term response developed. The greater power of the finite element analyses means that models can also be generated to analyze the experimental data, and greater fidelity can be added to the model as desired.

9.2.1 Interpretation of Cyclic Parallel Plate Data

The data from the cyclic parallel plate test will again consist of $tan\delta$ or G' and G''. The workstation code directly interprets the values for G' and $tan\delta$ at the frequency where G'' is a maximum. If a relative maximum cannot be found then a frequency should be chosen that is close to the loading frequency for the simulations. This will ensure that the correct response will result for the loading frequency.

9.2.2 Workstation Code Modeling

Once the properties have been determined all that remains is the data for the O-ring and gland geometry and the loading. The geometric inputs are readily determined from the proposed design. Face seals can also be modeled, which can be an aid in evaluating the cyclic parallel plate tests. Variations in the design geometry are not as easily evaluated because of the increase in computational times, but this is balanced by the more accurate models that can be examined. The loading again consists of a mean squeeze and mean pressure, and variations about the mean. The frequency of the variations must also be determined. The workstation allows for a phase angle to be introduced between the squeeze variations and the pressure variations.

9.2.3 Long Term Response Workstation Prediction

If the long term response is desired then the appropriate relaxation time must be used. Again the long term relaxation time is difficult to determine from typical compression set data as discussed in Section 7.5. For accurate steady state estimates, the long time G' and G'' must be known, or equivalently, a long time relaxation test must be performed. Because of the greater generality of the finite element code, the models generated can be readily modified to include nonlinear response parameters, such as creep properties in equations (12) and/or (46).

10.0 CONCLUSIONS

Benchtop procedures capable of determining basic properties of new and aged seals were examined and developed. In conjunction with the test methods, analytical models were developed with the intent of predicting sealing properties of aged O-ring seals.

Two methods potentially capable of measuring the storage and loss moduli of O-rings under dynamic conditions were explored. One method was based on measuring multiple rebounds of a pendulum dropped against the O-ring. The second method was based on determining the cyclic force- displacement response of O-rings squeezed between two parallel plates. The O-rings were excited over a series of frequencies ranging from 1 to 100 Hz.

Two computational methods were developed for the analysis of data generated by these test methods. The first was designed to run on a PC, and predicted the deformation and forces for O-rings for a variety of loading conditions. The second was a workstations version that automatically ran either of two finite element codes (MARC or ANSYS.)

Both test methods proved capable of measuring the basic properties of the O-rings, with somewhat greater accuracy from the pendulum rebound method. Variations between different materials were determined. However, the aging method did not induce significant changes in each family of O-rings, preventing the successful development of a true life prediction method.

The rebound test equipment and method has been retained and will be used in future efforts to determine fundamental properties of elastomer samples with non-standard geometries. An application for a patent covering the test hardware, test method and analytical software has been submitted to the UTRC legal department for filing.

11.0 REFERENCES

1. Rivlin, J., "Large Deformations of Isotropic Materials VI. Further Results in the Theory of Torsion, Shear and Flexure," Philosophical Transactions, **A242**, pp 173-195, 1949.
2. Treloar, L.R.G., The Physics of Rubber Elasticity, Clarendon Press, Oxford, 1958.
3. Fung, Y.C., Foundations of Solid Mechanics, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1965.
4. NASA, "Fitting Prony Series to Data on Viscoelastic Materials," NASA Tech Briefs MFS-28937, 1985.
5. Gorofalo, F. Fundamentals of Creep in Metals, The MacMillan Company, New York, 1965.
6. Krempl, E., and V.V. Kallianpur "The Uniaxial Unloading Behavior of Two Engineering Alloys at Room Temperature," ASME Journal of Applied Mechanics, **52**, pp. 654-658, 1985.
7. Nishiguchi, T.L. Sham and E. Krempl, "A Finite Deformation Theory of Viscoplasticity Based on Overstress," ASME Journal of Applied Mechanics, **57**, pp 548-552, 1990.
8. Salita, M., "Simple Finite-Element Model of O-Ring Deformation and Activation During Squeeze and Pressurization," AIAA Journal of Propulsion and Power, **4**, pp. 497-511, 1988.
9. Salita, M., 'A Simple Finite-Element Model of O-Ring Deformation and Activation During Squeeze and Pressurization,' AIAA-87-1739, Presented at the AIAA/SAE/ASME Joint Propulsion Conference, San Diego, June 29-July 12, 1987.

APPENDIX A
VISUAL BASIC CODE USER'S MANUAL

The Visual Basic version can be used on most PC's. It provides the user with a simple interface that prompts the user for the data entries and execution.

Double click on the program "vboring.exe" and a screen will appear with the data entries run labeled. Enter a title to identify the case to be run. Proceed to the next data entry line and enter the gland dimensions as shown in Figure A.1. On the next line enter in order: the minor or cross-section diameter, the stretched inner diameter, followed by the unstretched inner diameter. Proceed to the next line and enter the Young's modulus (use three times the storage modulus), the relaxation time (which is the period for the frequency where the loss modulus is a maximum, or if that is not known the period for the loading frequency), and last enter the relaxation factor (which can be taken to be the long term shear modulus divided by the short term modulus.) At the end of this line enter the file name without an extension (i.e., the job ID.) Skip to the next line and enter the applied pressure, the back pressure and the coefficient of friction. Move down one line to enter the loading type. This is usually a 2 for sinusoidal loading. Type 3 is exponential decay, while type 1 is step loading, and type 0 is used for program tests. The next three entries on the line are the change in the pressure, the change in the gap and the frequency. In the last line enter the number of time steps and the time step.

Hit the execute button to run the program. This will first run the Fortran program "make_input" which uses the Visual Basic input and converts it to an input file for the modified O-ring deformation program developed by Salita (Ref. 8.) The output from the O-ring deformation is stored in a file with the job ID and an '.out' appended. Next the Visual Basic program executes the Fortran program "plotit.f." The program plotit will ask the user how often the deformed plots should be completed. An entry of 5 will plot every fifth step. Program plotit first presents the deformed plots through increasing time. Hit the X key to get to the next plot. The last three plots present in order: the footprint history, the normal force history on each wall, and, if friction is present, the last plot is the frictional forces on each wall. Hard copies (in the form of postscript files) cannot presently be made of the deformed plots. Footprint and force history plots can be made by hitting the H key. Only one plot can be made per run because of incompatibilities between the graphics subroutines and DOS. The user will have to close the DOS window to complete the run. If no hard copies are requested, then the program exits normally.

The O-ring deformation code was written in Fortran and translated to C, and the C code was compiled using Gnu C. Hence, if the O-ring deformation program exits with an error,(usually as a segmentation fault) then the most likely explanation is that an improper model was generated. Any messages will refer to the C code and it will not be possible to identify the exact source of the error. As an example, an O-ring that does not seal and has pressure applied will exit with an error. But since the error message refers to the C code that was generated from the Fortran code, it cannot be readily traced.

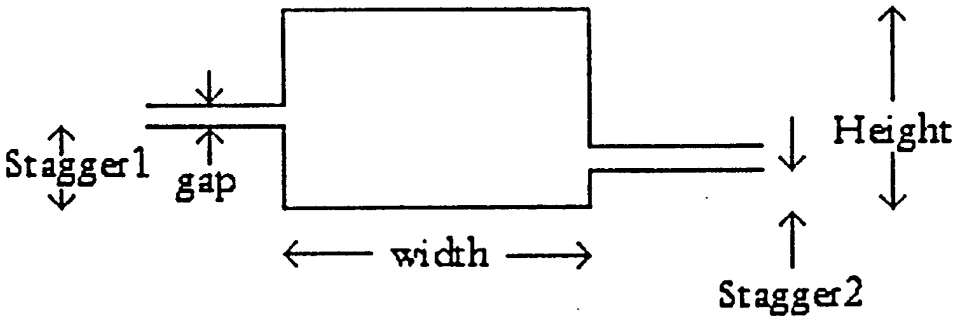


Fig. A.1 PC Code Gland Geomerty

APPENDIX B
DOS CODE USER'S MANUAL

The DOS version can be used on more PC's than the Visual Basic version, since it requires less computing capability, but it is also more difficult to use. On older machines the user first needs to exit windows, on machines with Windows '95 choose the MS-DOS prompt under the Programs menu.

Once in DOS type 'copy template.dat' followed by the job name. Append the job name with '.dat' and hit 'enter'. This copies a previous file to the new input data file. Next type 'edit' followed by the job name. Again append the job name with '.dat' and hit 'enter'. Once in the editor a form similar to Figure B.1 will appear on the screen. The data already entered is a previous test case used to check the program. Erase the text in the first line, by highlighting and then using cut under the edit theme, or by backspacing through the line. Enter a title that identifies the case to be run. Proceed down two lines and enter the gland dimensions as shown in Figure B.2 by erasing the previous entries and typing the appropriate quantities. Skip down another two lines and enter in order: the minor or cross-section diameter, the stretched inner diameter, followed by the unstretched inner diameter. Skip down two lines and enter the Young's modulus (use three times the storage modulus), the relaxation time (which is the period for the frequency where the loss modulus is a maximum, or if that is not known the period for the loading frequency), and last enter the relaxation factor (which can be taken to be the long term shear modulus divided by the short term modulus.) Skip another line and on the next line enter the applied pressure, the back pressure and the coefficient of friction. Move down two lines to enter the loading type. This is usually a 2 for sinusoidal loading. Type 3 the is exponential decay. Type 1 is step loading, and type 0 is used for program tests. The next three entries on the line are the change in the pressure, the change in the gap and the frequency. In the last line enter the number of time steps and the time step.

Exit the editor by entering the File menu, choosing save, and then re-enter the File menu and choose exit. At the DOS prompt, type "runoring" followed by the job name. This will execute the file runoring.bat, which is shown in Figure B.3. This file will first run the compiled Fortran program "make_input" (see the listing after this appendix) which uses the file from the editor and converts it to an input file for the modified O-ring deformation program developed by Salita (Ref. 8.) The output from the O-ring deformation is stored in a file with the job name and an '.out' appended. Runoring.bat next stores the job name in jobid.tmp and then executes the program Fortran program plotit.f (see the listing at the end of this appendix). The program plotit will ask the user for how often the deformed plots should be completed. An entry of 5 will plot every fifth step. Program plotit first presents the deformed plots through increasing time. Hit the X key to get to the next plot. The last three plots present in order: the footprint history, the normal force history on each wall, and if friction is present, the last plot is the frictional forces on each wall. Hard copies (in the form of postscript files) cannot presently be made of the deformed plots. Footprint and force history plots can be made by hitting the H key. Only one plot can be made per run because of incompatibilities between the graphics subroutines and DOS. The user will have to close the DOS window to complete the run. The last step in runoring.bat is to delete the jobid.tmp file.

After completing a run it is possible to replot the results since the output has been stored. At the DOS prompt, type plotit. The program will ask for the job name and then ask for the plot frequency as described above. A postscript file of one of the history plots can be obtained here also.

The O-ring deformation was written in Fortran and translated to C, and the C code was compiled using Gnu C. Hence, if the O-ring deformation program exits with an error,(usually as a segmentation fault when an improper model is generated), then any messages will refer to the C code and it will not be possible to identify the exact source of the error. As an example, an O-ring that does not seal and has pressure applied will exit with an error. But since the error message refers to the C code that was generated from the Fortran code, it cannot be readily traced.

```
Compression Set Test Case
   gap   width   height  stagger_l   stagger_2
  .003    .193    .1205       0.         0.
 minor_dia   major_inner_dia   unstrech_dia
    .139            0.998          0.984
    E   relax_time   relax_fac
  1200.         2.        0.7
  pext   pback     mu
  900.     15.     .0
 type   d_press   d_gap   period
   2        0.       0.      5.0
 no_time   d_time
    10      1.0
```

Figure B.1

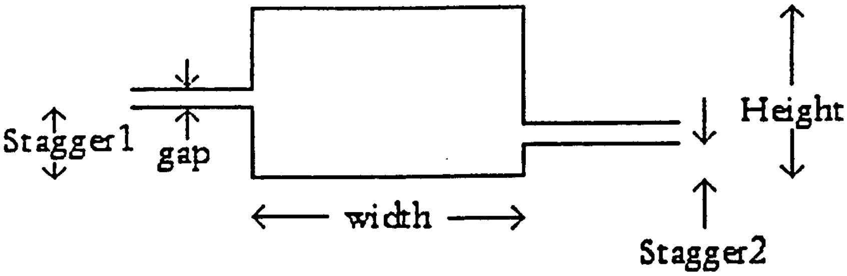


Figure B.2 PC Code Gland Geometry

```
make_input < $1.dat > $1.inp
oringdef < $1.inp > $1.out
echo $1 > jobid.tmp
plotit
rm jobid.tmp
```

Figure B.3

```
      program make_input
C
C  Make the input file to run oringdef
C
      implicit none
C
C  Declare variables
C
      character*79 Title                  ! Case title
      character*79 dummy(10)              ! Dummy characters
      real gap                            ! initial gap
      real width                          ! Width
      real height                         ! Height
      real minor_dia                      ! Minor diameter
      real major_inner_dia                ! Indside major diameter
      real unstrech_dia                   ! Unstretched major diameter
      real stagger_1                      ! Stagger 1 dimension
      real stagger_2                      ! Stagger 2 dimension
      real pext                           ! External pressure
      real E                              ! Young's modulus
      real relax_time                     ! Relaxation constant
      real mu                             ! Coefficient of friction
      real relax_fac                      ! Realxation factor
      real pback                          ! Back pressure
      integer no_time                     ! Number of time steps
      real d_press                        ! Variable pressure amplitude
      real d_gap                          ! Variable gap amplitude
      real d_time                         ! time step
      real period                         ! Period
      integer type                        ! Time functional type
C
C  Get data
C
      read (5,'(A)') Title
      read (5,'(A)') dummy(1)          ! Heading line
      read (5,*) gap,width,height,stagger_1,stagger_2
      read (5,'(A)') dummy(1)          ! Heading line
      read (5,*) minor_dia,major_inner_dia,unstrech_dia
      read (5,'(A)') dummy(1)          ! Heading line
      read (5,*) E,relax_time,relax_fac
      read (5,'(A)') dummy(1)          ! Heading line
      read (5,*) pext,pback,mu
      read (5,'(A)') dummy(1)          ! Heading line
      read (5,*) type,d_press,d_gap,period
      read (5,'(A)') dummy(1)          ! Heading line
      read (5,*) no_time,d_time
```

```
C
C  Write input file
C
      write (6,'(79A)') Title
      write (6,'(A1)') '7'
      dummy(1)='   .00'
      write (6,'(1P6E10.3,A6 / 1P2E10.3)')
     $    gap,width,height,minor_dia,major_inner_dia
     $   ,unstrech_dia,dummy(1),stagger_1,stagger_2
      dummy(2)='  .000'
      write (6,'(1P4E10.3,A6,1PE10.3,A6,1PE10.3)')
     $    pext,-E,relax_time,mu,dummy(1),relax_fac,dummy(1),pback
      dummy(3)='    36'
      dummy(4)='   101     1     0     1     1     0     0     1'
      write(6,'(A6,I6,A48)') dummy(3),no_time,dummy(4)
      write(6,'(1P3E10.3,I5,1PE10.3)')
     $    d_press,d_gap,d_time,type,period
      dummy(5)=' '
      write(6,'(A6)') dummy(5)
C
C  Normal end
C
      stop
      end
      program plotit
C
C  Plot the shape of the oring output
C
      implicit none
C
C  Declare geometric variables
C
      real gap                          ! Initial joint gap
      real width                        ! Gap width
      real height                       ! Gap height
      real Odia                         ! O-ring diameter
      real Rjoint                       ! Joint radius
      real Rorg                         ! Original oring diameter
      real g_rate                       ! Joint growth rate
      real stag_1                       ! Forward joint stagger
      real stag_2                       ! Rearward joint stagger
      real pext                         ! External pressure
      real time_now                     ! Current time
      real old_time                     ! Last time read
      integer loop                      ! Loop number
      integer old_loop                  ! Old loop number
C
C  Declare O-ring finite element variables
C
      real zj(37)                       ! Current axial position
      real rj(37)                       ! Current radial position
      real zo(37)                       ! Original axial position
      real ro(37)                       ! Original radial position
      real wj(37)                       ! Axial displacement
      real uj(37)                       ! Radial displacement
      real foot(4,1001)                 ! Footprints
      real norm(4,1001)                 ! Normal force
      real fric(4,1001)                 ! Friction force
      real time(1001)                   ! Time
      real plot_time_interval           ! TIme interval between plots
      real dtime                        ! TIme interval between output
      integer time_step                 ! TIme steps between plots
      integer No_time_steps             ! Number of time steps
      integer i,k                       ! Counters
      integer EOF                       ! End of file flag
      character*4 iter                  ! Iteration number
      character*138 dum                 ! Dummy line
C
C  Declare Plot variable
C
      character*40 Title             ! First title line
C
C  Declare job ID variables
C
      character*40 jobid                        ! Job ID
      character*45 infile                       ! Input file name
      character*45 outfile                      ! Output file name
      integer jobid_length                      ! Length of jobid string
      integer string_length                     ! String length function
```

```
C
C  Initialize Jobid to blanks
C
      do i=1,40
         jobid(i:i)=' '
      end do
C
C  Get time step between plots
C
      plot_time_interval=0
      write (6,'("Please enter time step between geometry plots: ",$)')
      read (5,*) plot_time_interval
      if (plot_time_interval.le.0) plot_time_interval=1
      old_time=0.
      No_time_steps=0
C
C  Get jobid
C
      open (unit=31,file='jobid.tmp',status='unknown')
      EOF=0
      read (31,'(A)',iostat=EOF) jobid
      if (EOF.eq.0) then
         jobid_length=string_length(jobid,40)
      else if (EOF.ne.0.or.jobid_length.eq.0) then
         write (6,'("Please enter job id: ",$)')
         read (5,'(A)') jobid
         jobid_length=string_length(jobid,40)
      end if
C
C  Store input & output file names
C
      infile=jobid(1:jobid_length)//'.inp'
      outfile=jobid(1:jobid_length)//'.out'
C
C  Open the input and outputfiles
C
      open (unit=20,file=infile(1:jobid_length+4),status='old')
      open (unit=21,file=outfile(1:jobid_length+4),status='old')
C
C  Read the job title
C
      read (20,'(A40)') Title
C
C  Read the geometric quantities
C
      read (20,'(A)') dum
      read (20,*) gap,width,height,Odia,Rjoint,
     $              Rorg,g_rate,stag_1,stag_2
C
C  Read the coordinates for the first solution
C
      EOF=0
      old_loop=0
      DO while (EOF.eq.0)
      k=0
      do while (k.eq.0)
         read (21,'(A90)',IOSTAT=EOF) dum
         if (EOF.ne.0) then
```

```
C
C  Normal program end
C
            call history_plots(foot,norm,fric,time,No_time_steps,Title)
            call grclos
            close(unit=20)
            close(unit=21)
            stop
         end if
C
C  Get deformed shape
C
         if (dum(11:60).eq.
     $      'SOLUTION OF DISPLACEMENT EQS USING GAUSS-JORDAN...'
     $      .and.old_loop.ne.loop) k=1
         if (dum(11:23).eq.'ITERATION # =') then
            iter=dum(24:27)
         end if
C
C  Get Subtitle parameters
C
         if (dum(6:10).eq.'LOOP=')then
            read (dum,'(10X,I5,15X,F8.0,13X,F7.0,15X,F10.0)')
     $         loop,pext,gap,time_now
               if (time_now.ne.0..and.old_time.le.0.) then
                  dtime=time_now
                  time_step=plot_time_interval/dtime+.1
                  old_time=time_now
               end if
               No_time_steps=No_time_steps+1
               time(No_time_steps)=time_now
         end if
C
C  Get footprint and force history
C
         if (dum(11:39).eq.'FOOTPRINTS AND WALL FORCES...') then
            read (21,'(A90)') dum
            do i=1,4
               read (21,'(37X,3F15.0)') foot(i,No_time_steps)
     $                                 ,norm(i,No_time_steps)
     $                                 ,fric(i,No_time_steps)
            end do
         end if
      end do
      read (21,'(A10)') dum
      do i=1,37
         read (21,'(19X,4F15.0)') uj(i),wj(i),rj(i),zj(i)
      end do
      old_loop=loop
C
C  Make a deformed plot at the appropriate time steps
C
      if (time_step.eq.0) then
         call deformed_plot(gap,pext,time_now,rj,uj,zj,wj
     $                     ,stag_1,width,height,stag_2,Title)
```

```
      else if (((No_time_steps-1)/time_step)*time_step
     $          .eq.No_time_steps-1) then
         call deformed_plot(gap,pext,time_now,rj,uj,zj,wj
     $                     ,stag_1,width,height,stag_2,Title)
      end if
      END DO
      end


      subroutine history_plots(foot,norm,fric,time,No_time_steps,Title)
C
C  Plot the summary history plots of the footprints and forces
C
      implicit none
C
C  Declare input variables
C
      real foot(4,1001)              ! Footprints
      real norm(4,1001)              ! Normal force
      real fric(4,1001)              ! Friction force
      real time(1001)                ! Time
      integer No_time_steps          ! Number of time steps
C
C  Declare subroutine variables
C
      integer i,j,k                ! Counters
      real sumsqry                 ! Test varaible for friction history plot
C
C  Declare plot variables
C
      real x(2002)                 ! String array of x values
      real y(2002)                 ! String array of y values
      character*40 Title           ! First title line
      character*40 Subtitle        ! Second title line
      character*11 Subtitle1       ! Second title line - part 1
      character*13 Subtitle2       ! Second title line - part 2
      character*16 Subtitle3       ! Second title line - part 3
      character*18 Xtitle          ! X axis title
      character*18 Ytitle          ! Y axis title
      character*79 pltitl          ! String containg title information
      character*10 titl(30)        ! Line title
      integer Ilin(30)             ! Line type
      integer Isym(30)             ! Symbol type
      integer points(30)           ! No. of points for each line
      integer nline                ! No. of lines
C
C  Set plot title & initialize plots
C
      call grinit(5,6,Title)
      Xtitle='Time'
      nline=4
      titl(1)='Inside'
      titl(2)='Back'
      titl(3)='Outside'
      titl(4)=’Pressure’
      do i=1,4
         ilin(i)=i
         isym(i)=0
         points(i)=No_time_steps
      end do
C
C  Make the footprint time history plot
C
      Ytitle=’Footprint’
      Subtitle=’Footprint History’
      k=0
      do j=1,4
         do i=1,No_time_steps
              k=k+1
              x(k)=time(i)
              y(k)=foot(j,i)
         end do
      end do
      pltitl=’~’//Xtitle//’~’//Ytitle//’~’//Subtitle
      call set_hard_copy
      call set_original
      call grklin(Ilin,Isym,points,titl,nline,x,y,pltitl,84)
C
C  Make normal force history plot
C
      Ytitle=’Normal Force’
      Subtitle=’Normal Force History’
      k=0
      do j=1,4
         do i=1,No_time_steps
              k=k+1
              x(k)=time(i)
              y(k)=norm(j,i)
         end do
      end do
      pltitl=’~’//Xtitle//’~’//Ytitle//’~’//Subtitle
      call set_hard_copy
      call set_original
      call grklin(Ilin,Isym,points,titl,nline,x,y,pltitl,84)
C
C  Make friction force history plot
C
      sumsqry=0.
      Ytitle=’Friction Force’
      Subtitle=’Friction Force History’
      k=0
      do j=1,4
         do i=1,No_time_steps
              k=k+1
              x(k)=time(i)
              y(k)=fric(j,i)
              sumsqry=sumsqry+y(k)**2
         end do
      end do
      if (sumsqry.lt.1.e-6) return  ! Test for no friction forces
      pltitl='~'//Xtitle//'~'//Ytitle//'~'//Subtitle
      call set_hard_copy
      call set_original
      call grklin(Ilin,Isym,points,titl,nline,x,y,pltitl,84)
C
C  Normal return
C
      return
      end


      subroutine deformed_plot(gap,pext,time_now,rj,uj,zj,wj
     $                      ,stag_1,width,height,stag_2,Title)
C
C  Declare input variables
C
      real gap                       ! Initial joint gap
      real pext                      ! External pressure
      real time_now                  ! Current time
      real rj(37)                    ! Current radial position
      real uj(37)                    ! Radial displacement
      real zj(37)                    ! Current axial position
      real wj(37)                    ! Axial displacement
      real stag_1                    ! Forward joint stagger
      real width                     ! Gap width
      real height                    ! Gap height
      real stag_2                    ! Rearward joint stagger
C
C  Declare temporary variables
C
      real r_inner(6),z_inner(6)     ! Inner wall node locations
      real r_outer(6),z_outer(6)     ! Outer wall node locations
      real smin                      ! Minimum coordiante
      real smax                      ! Maximum coordiante
C
C  Declare plot variables
C
      real x(2002)                ! String array of x values
      real y(2002)                ! String array of y values
      character*40 Title          ! First title line
      character*40 Subtitle       ! Second title line
      character*11 Subtitle1      ! Second title line - part 1
      character*13 Subtitle2      ! Second title line - part 2
      character*16 Subtitle3      ! Second title line - part 3
      character*18 Xtitle         ! X axis title
      character*18 Ytitle         ! Y axis title
      character*79 pltitl         ! String containg title information
      character*10 titl(30)       ! Line title
      integer Ilin(30)            ! Line type
      integer Isym(30)            ! Symbol type
      integer points(30)          ! No. of points for each line
      integer nline               ! No. of lines
      integer i,k                    ! Counters
C
C  Set plot titles & initialize plots
C
      Xtitle=' '
      write (Subtitle1,'(A,F7.4)') 'gap=',gap
      write (Subtitle2,'(A,F7.1)') ' Pext=',pext
      write (Subtitle3,'(A,1PE10.3)') ' Time=',time_now
      Subtitle=Subtitle1//Subtitle2//Subtitle3
      Ytitle=' '
      call grinit(5,6,Title)
C
C  Make plot array for current oring cross-section
C
      do i=1,36
         x(i)=rj(i+1)
         y(i)=zj(i+1)
      end do
      x(37)=x(1)
      y(37)=y(1)
      points(1)=37
      ilin(1)=1
      isym(1)=0
      titl(1)='Current'
C
C  Make plot array for original oring cross-section
C
      do i=38,73
         x(i)=rj(i-36)+uj(i-36)
         y(i)=zj(i-36)-wj(i-36)
      end do
      x(74)=x(38)
      y(74)=y(38)
      points(2)=37
      ilin(2)=2
      isym(2)=0
      titl(2)='Original'
C
C  Make inner walls
C
      r_outer(1)=stag_1
      z_outer(1)=.4+width
      r_outer(2)=stag_1
      z_outer(2)=.2+width
      r_outer(3)=height
      z_outer(3)=.2+width
      r_outer(4)=height
      z_outer(4)=.2
      r_outer(5)=stag_2
      z_outer(5)=.2
      r_outer(6)=stag_2
      z_outer(6)=0.
      k=76
      do i=1,6
```

```
         k=k+1
         x(k)=r_outer(i)
         y(k)=z_outer(i)
      end do
      points(4)=6
      ilin(4)=3
      isym(4)=0
      titl(4)='Outer wall '
c
c  Make outer walls
c
      r_inner(1)=stag_1-gap
      z_inner(1)=.4+width
      r_inner(2)=stag_1-gap
      z_inner(2)=.2+width
      r_inner(3)=-gap
      z_inner(3)=.2+width
      r_inner(4)=-gap
      z_inner(4)=.2
      r_inner(5)=stag_2-gap
      z_inner(5)=.2
      r_inner(6)=stag_2-gap
      z_inner(6)=0.
      do i=1,6
         k=k+1
         x(k)=r_inner(i)
         y(k)=z_inner(i)
      end do
      points(5)=6
      ilin(5)=3
      isym(5)=0
      titl(5)='Inner wall '
c
c  Make plot array for square array
c
      smin=1.E10
      smax=-1.E10
      do i=1,k
         if (i.ne.75.and.i.ne.76) then
            if (smin.gt.x(i)) smin=x(i)
            if (smin.gt.y(i)) smin=y(i)
            if (smax.lt.x(i)) smax=x(i)
            if (smax.lt.y(i)) smax=y(i)
         end if
      end do
      x(75)=smin
      y(75)=smin
      x(76)=smax
      y(76)=smax
      points(3)=2
      ilin(3)=0
      isym(3)=0
      titl(3)=' '
```

```
C
C  Make plots
C
      nline=5
      pltitl='~'//Xtitle//'~'//Ytitle//'~'//Subtitle
      call set_hard_copy
      call grklin(Ilin,Isym,points,titl,nline,x,y,pltitl,84)
C
C  Normal return
C
C     call grclos
      return
      end


      integer function string_length(string,max_length)
C
C  Return length of character string
C
      implicit none
C
C  Declare variables
C
      character*80 string                 ! Character string
      integer max_length                  ! Maximum length of character string
C
C  Initialize string length
C
      string_length=0
C
C  Check to be sure string length does not exceed declaration
C
      if (max_length.gt.80) then
         write(6,*) '***********************************************'
         write(6,*) '*  Max_length in function string_length > 40  *'
         write(6,*) '***********************************************'
         return
      end if
C
C  Check for a null string
C
      if (string.eq.'') return
C
C  Find string length
C
      do while (string(string_length+1:string_length+1).ne.' '
     $           .and.string_length.lt.40)
         string_length=string_length+1
      end do
C
C  normal return
C
      return
      end
      subroutine set_hard_copy
c
c  Set the hard copy to produce multiple copies
c
      implicit none
c
c  Declare variable
c
      character*40 script(40)
      script(1)='Q'
      script(2)='   10'
      script(3)='    0'
      script(4)='I'
      call grscpt(4,script)
c
c  Normal return
c
      return
      end


      subroutine set_original
c
c  Set to scaled 'original' size
c
      implicit none
c
c  Declare variable
c
      character*40 script(40)
      script(1)='O'
      script(2)='I'
      call grscpt(2,script)
c
c  Normal return
c
      return
      end
```

APPENDIX C
MARC MODELING USER'S MANUAL

MARC workstation modeling provides the experienced user of finite element codes with a significant increase in capability. Users with less experience may want to simulate tests that cannot be completed with the simpler PC versions.

The first step is to write a file containing the O-ring geometry, material properties, and loading conditions, as shown in Fig C.1. A job name must be chosen, and can consist of any number of characters. It must be appended with an '.inp'. The first line of the file is a title.

The second line contains the major and minor diameter. The major diameter is the inside diameter plus the cross-section radius. The minor diameter is the cross-section diameter. Line 3 contains the storage modulus and line 4 contains *tanδ.* Line 5 is the frequency at which lines 3 and 4 apply. If line 5 is the frequency at which the *tanδ* is a maximum, the accuracy at other frequencies will be significantly increased. Line 6 contains the friction coefficient.

The gland geometry now follows. Line 7 contains the type of gland (type 1 is a piston seal, type 2 is a rod seal and type 3 is a face seal, as shown in Figure C.2.) The gland width and depth are entered on line 8. For a definition of the geometric dimensions see Figure C.3. On line 9 the inner diameter and the initial gap are entered . If zeros are entered for the quantities on line 9 then software will automatically assign values to each so that the O-ring just touches the inside of the gland and the top of the O-ring just touches the gland.

The loads are entered next. Line 10 contains the internal and external pressure. Line 11 must contain the mean squeeze, and line 12 the maximum squeeze. Line 13 has the pressure change, the phase angle with respect to the gland motion. On line 14 the number of cycles to be simulated, and the frequency are entered. The last line, line 16, contains information on the number of increments required to apply the mean squeeze, then the mean pressure, and lastly the cyclic loads.

The user can now run the simulation by typing "run_oring" followed by the job name. The script run_oring is included as Figure C.4. The first line echoes the job name to the ACSII file jobname.txt. Next the compiled Fortran program "make_model" is run which generates the MARC input data. The file jobname.txt is no longer needed and is next removed. Two subroutines, that describe the pressure (forcem.f) and gland motion (motion.f) history needed for the MARC finite element code, are now stored in a file with the job name followed by '.f'. Now the MARC program can be run. The program is automatically brought up, told the data file name (generated by the Fortran program "make_model.f") and told to run in background mode. The question asked by the MARC shell script is answered. If the job name is incorrect the MARC shell script will continue to ask for an answer, and the user must terminate it by holding down the Control key and the C key. The last line of the script deletes the file that contains the subroutines forcem.f and motion.f.

During the run the MARC code will store a log of the run in a file with job name appended by '.log'. Output is written in the file with the job name appended by '.out', and data for post processing is in the file with the job name followed by '.t16'.

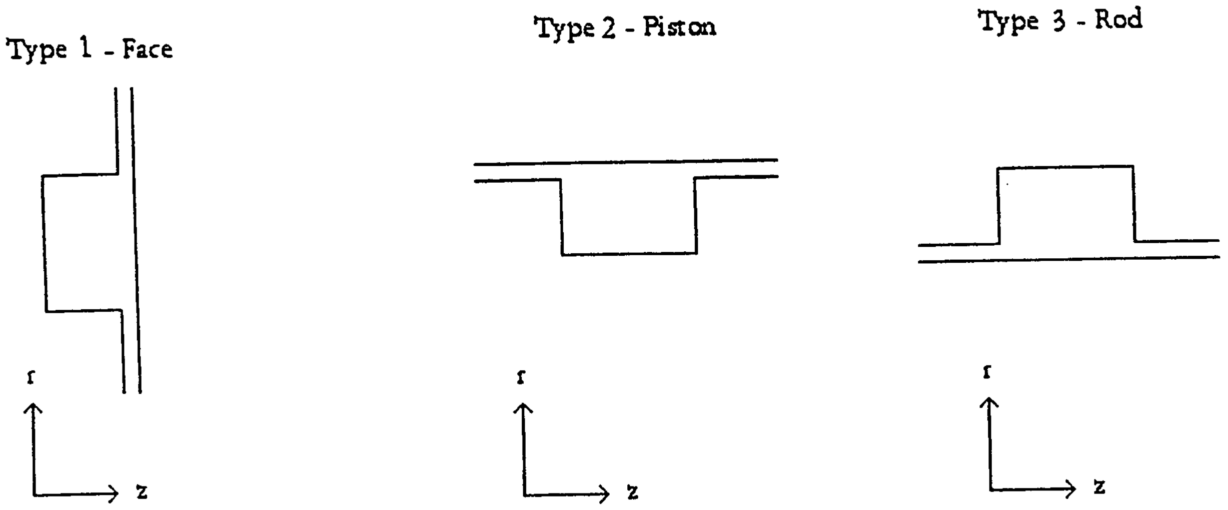
Listings of the subroutines forcem.f, and motion.f are attached at the end of this appendix. The subroutines have three basic sections. The first programs the squeeze. The second brings the pressures to the mean pressures, and the last performs the cyclic loading. The listing for make_model.f follows. It generates each data block for the MARC input file. Many of the lines are the same for each run, and some of these are stored in several files appended with '.inp'. The finite element mesh is also stored for a unit cross-section radius. The mesh is translated according to the major radius, and rotated to accommodate the three O-ring types. There is an additional rotation of 45 degrees to minimize the effects of mesh distortion on the results.

A summary of the user input data is sent to the screen and also to a file with the job name appended with a '.sum'. A list of files used by make_model.f is included as Figure C.5.

During the MARC runs, one persistent error occurs. At high pressure loadings the simulation detects instabilities (either buckling or numerical) and will terminate with one or more messages indicating that the third invariant of particular elements is negative. This indicates that the loading increments (or total load) cannot be accommodated without turning elements inside out. Although this is an indication of a true physical instability, it is more likely to be numerical in nature.

```
Test case for oring contact
.25,.0625                       ! Major diameter, minor diameter
2.e3                            ! Storage modulus
0.15                            ! Tan(delta) max. @ frequency below
15.                             ! Frequency
0.102                           ! Friction coefficient
3                               ! 1-Face, 2-Piston, 3-Rod
.075,.0375                      ! Gland width, depth
0.,0.                           ! Inner edge diameter, gap
25.,15.                         ! Internal, external pressure
.15                             ! Mean compression squeeze
.20                             ! Maximum compression squeeze
15.,0.                          ! Internal pressure change & phase angle
1,15.                           ! No. of cycles & frequency of oscillations
10,1,40                        ! No. of inc's: squeeze, pressure, per cycle
```

Figure C.1



Type 1 - Face

Type 2 - Piston

Type 3 - Rod

Figure C.2 Gland Types

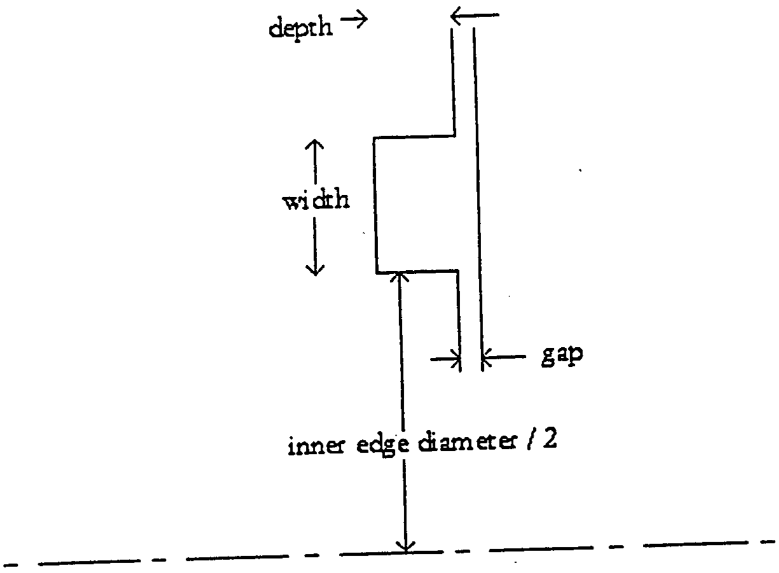


Figure C.3 Gland Geometry

```
echo $1 > jobname.txt
make_model < $1.inp
rm jobname.txt
cat forcem.f motion.f > $1.f
/utrc/home/seh/marcment/marck62/tools/run_marc -j $1 -u $1 -q b < y.inp
rm $1.f
```

Figure C.4

File	Description
coor.inp	Finite element mesh coordinates
conn.inp	Finite element mesh connectivity
head.inp	Parameter & model definition heading
press.inp	Pressure element sets
endopt.inp	End of model definition
control.inp	Control data block
dist.inp	Typical distributed load data block

Figure C.5

```
      subroutine forcem(press,th1,th2,nn,n)                            dp
      implicit real*8 (a-h,o-z)
      dimension n(7)
      integer type
      logical ext,int,even,odd,contact
      common/blnk/sigte(33),epte(33),f(40),f1(40),sum(40),sumd(40),
     1frx(40),dx(40),instab,lm(30)
      common/dimen/nstrmx,nnodmx,nno2mx,nbctmx,idss,nqnp,itie,numel,
     1numbc,nuxtr,ngans,maxnp,numnp,ndeg,mprmax,ncrd,nsxx,itiem,
     2istypm,longsm,longtm,maxnpr,maxqnp
     1,neltyp,maxser,n14siz,maxall, ninert,nstra,nstram
     4,nintb,maxavm,nstavm,nintpv,ndegmx,ncrdmx,ngenmx,nintps
     5,nusdat
      common/array2/iex,isigxx,iitype,inpi,ieplas,iitmat,intmat,inpmat,
     1inpbt,idisp,idispt,ixord,inp,idsx,idsx2,idsx1,idsxt,itx,ipinc,
     2iptot,ixload,imaxco,inap,impres,iequiv,istint,istvar,idsbuk,
     3inaprh,icoord,idump,ietota,idump1,llpos,igsig,iepl
     1,iswell,isera,isern,ieelas,iecorr,icauch
     5,isigco,isigsi,isiggc,isiggs,iepsco,iepssi,iepsgc,iepsgs
     6,ijmprs,itx2,itytra,isigxl,ieprat,idsxr,ietot3,idsbuc,idispc
     7,jexs,itx3,iex3
      common /brice/ old_inc,old_time,old_dtime
      common/space/ints(1)
      dimension vars(1)
      equivalence(ints(1),vars(1))
      dimension ccnode(12),ddnode(12),x1(4),x2(4),dx1(4),dx2(4)
c* * * * * *
c
c     defined non-uniformed distributed force on an element.
c
c     press        distributed load increment magnitude
c     th1          coordinate
c     th2          coordinate
c     nn           integration point number
c     n            element number,etc
c
c* * * * * *
C
C  Read the times from the call to motion
C
      open (unit=132,file='times.inp',status='old')
      read (132,*) inc,time,dtime
      close (unit=132)
C
C  Get the date for the transient motion
C
      open (unit=21,file='temp.inp',status='unknown')
      read (21,*) delta
      read (21,*) d_delta
      read (21,*) pint
      read (21,*) pext
      read (21,*) dp
      read (21,*) omega
      read (21,*) phi
      read (21,*) type
      read (21,*) d_dia
      read (21,*) No_squeeze
      read (21,*) No_press
      close(unit=21)
C
C  Write the input to a fortran file
C
      if ((n(1).eq.1.or.n(1).eq.18).and.nn.eq.1) then
         write (77,'(A,I5,2(A,1PE10.3))')
     $   'Input: inc = ',inc ,'   time = ',time ,'   dtime = ',dtime
         write (77,'(2(A,I5))')
     $   'Input: No_press = ',No_press,'   No_squeeze = ',No_squeeze
      end if
C
C  Determine if the element is an internal or external pressure element
C
      int=.false.
      ext=.false.
      int=int.or.(n(1).ge.1.and.n(1).le.18)
      int=int.or.(n(1).ge.1261.and.n(1).le.1278)
      ext=ext.or.(n(1).ge.19.and.n(1).le.36)
      ext=ext.or.(n(1).ge.1279.and.n(1).le.1296)
      even=(n(1)/2)*2.eq.n(1)
      odd=.not.even
      int=int.or.((.not.int).and.(.not.ext).and.odd)
      ext=.not.int
      if (inc.eq.0) then
         write (88,*) 'n(1),odd,even,ext,int',n(1),odd,even,ext,int
      end if
C
C  Get the coordinates & displacements and store total
C
      if (int.or.ext) then
         do i=1,4
            lint=lm(i)
            jrdpre=0
            call vecftc(ccnode,vars(ixord),ncrdmx,ncrd,lint,jrdpre,2,1)
            jrdpre=0
            call vecftc(ddnode,vars(idsxt),ndegmx,ncrd,lint,jrdpre,2,5)
            x1(i)=ccnode(1)+ddnode(1)
            x2(i)=ccnode(2)+ddnode(2)
         end do
C
C  Determine if this element is in contact with a rigid surface
C
         contact=.false.
         do i=1,4
            if (i.gt.1) then
               dx1(i)=x1(i)-x1(i-1)
               dx2(i)=x2(i)-x2(i-1)
            else
               dx1(1)=x1(1)-x1(4)
               dx2(1)=x2(1)-x2(4)
            end if
            if (dx1(i)**2.lt.1.e-11) contact=.true.
            if (dx2(i)**2.lt.1.e-11) contact=.true.
         end do
      end if
      write (99,*) 'inc = ',inc,' n = ',n(1),'  contact = ',contact
C
C  Squeeze the O-ring without adding pressure
C
      if (inc.le.No_squeeze) then
         press=0.
         if ((n(1).eq.1.or.n(1).eq.18).and.nn.eq.1) then
            write (77,'(A,1PE10.3,2(A,I5))')
     $      'Output: press = ',press,'   ele. ',n(1),'   pt. ',nn
            write (77,'(/)')
         end if
         return
      end if
C
C  Linearily increase the pressure after initial squeeze
C
      if (inc.gt.No_squeeze.and.inc.le.No_squeeze+No_press) then
         if (int) then
            press=(pint+dp*sin(phi))

     $            *(inc-No_squeeze)**2/No_press**2
         else
            press=pext
     $            *(inc-No_squeeze)**2/No_press**2
         end if
         if (contact) press=0.
      end if
C
C  Find the pressure during the cyclic loading
C
      if (inc.gt.No_squeeze+No_press) then
         if (int) then
            press=pint+dp*sin(omega*time+phi)
         else
            press=pext
         end if
         if (contact) press=0.
      end if
C
C  Write the input to a fortran file
C
      if ((n(1).eq.1.or.n(1).eq.18).and.nn.eq.1) then
         write (77,'(A,1PE10.3,2(A,I5))')
     $   'Output: press = ',press,'   ele. ',n(1),'   pt. ',nn
         write (77,'(/)')
      end if
C
C  Normal return
C
      return
      end
      subroutine motion(x,f,v,time,dtime,nsurf,inc)
c
c
c * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c
c     user routine to provide surface motion data
c
c 2-d:
c     input  :   nsurf     - number of surface for which data is
c                            requested
c                time      - time at which data is requested
c                dtime     - current time increment
c                x(3)      - current die defining coordinates
c                            x(1) = 1st coordinate of center of
c                                   rotation
c                            x(2) = 2nd coordinate of center of
c                                   rotation
c                            x(3) = angle rotated around z-axis
c
c                f(3)      - current surface load
c                            f(1) = 1st component of load
c                            f(2) = 2nd component of load
c                            f(3) = moment
c                inc       - increment number
c
c     output :   v(3)      - current surface velocities
c                            v(1) = 1st component of center of
c                                   rotation velocity
c                            v(2) = 2nd component of center of
c                                   rotation velocity
c                            v(3) = angular velocity
c
c
c 3-d:
c     input  :   nsurf     - number of surface for which data is
c                            requested
c                time      - time at which data is requested
c                dtime     - current time increment
c                x(6)      - current die defining coordinates
c                            x(1) = 1st coordinate of center of
c                                   rotation
c                            x(2) = 2nd coordinate of center of
c                                   rotation
c                            x(3) = 3rd coordinate of center of
c                                   rotation
c                            x(4) = 1st component of direction cosine
c                            x(5) = 2nd component of direction cosine
c                            x(6) = 3rd component of direction cosine
c
c                f(6)      - current surface load
c                            f(1) = 1st component of load
c                            f(2) = 2nd component of load
c                            f(3) = 3nd component of load
c                            f(4) = 1st component of moment
c                            f(5) = 2nd component of moment
c                            f(6) = 3rd component of moment
c                inc       - increment number
c
```

```
c
c     output :   v(4)       - current surface velocities
c                             v(1) = 1st component of center of
c                                    rotation velocity
c                             v(2) = 2nd component of center of
c                                    rotation velocity
c                             v(3) = 3nd component of center of
c                                    rotation velocity
c                             v(4) = angular velocity
c
c
c* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c
c
      implicit real*8 (a-h,o-z)                                        dp
      dimension x(1),v(1),f(1)
      integer type
C
C  Write the input data to a fortran file for a check
C
      write (66,'(A,I5,2(A,1PE10.3))')
     $  'Motion: inc = ',inc,' time = ',time,' dtime = ',dtime
C
C  Write the current time & dtime to a file for use by forcem
C
      open (unit=132,file='times.inp',status='old')
      write (132,'(I5,1P2E15.4)') inc,time,dtime
      close (unit=132)
C
C  Get the date for the transient motion
C
      open (unit=21,file='temp.inp',status='unknown')
      read (21,*) delta
      read (21,*) d_delta
      read (21,*) pint
      read (21,*) pext
      read (21,*) dp
      read (21,*) omega
      read (21,*) phi
      read (21,*) type
      read (21,*) d_dia
      read (21,*) No_squeeze
      read (21,*) No_press
      close(unit=21)
C
C  Write the input to a file
C
      write (66,'(2(A,I5),A,1PE10.3)')
     $  'Input:  type ',type,' surf ',nsurf,'  d_dia = ',d_dia
      write (66,'(3(A,1PE10.3))')
     $  'Read:  delta = ',delta,' d_delta = ',d_delta,' omega = ',omega
```

```
c
c  Velocities are zero unless the conditions are right
c
      v(1)=0.
      v(2)=0.
      v(3)=0.
c
c  For increment zero return zero motion
c
      if (inc.eq.0) then
         write (66,'(3(A,1PE10.3),A)')
     $   'Output: v1 = ',v(1),' v2 = ',v(2),' v3 = ',v(3),' : Case 0'
         write (66,'(/)')
         return
      end if
c
c  Stretch the O-ring initially
c
      if (inc.le.No_squeeze.and.type.eq.3
     $   .and.d_dia.gt.0..and.nsurf.eq.3) then
         v(1)=0.
         v(2)=d_dia/float(2*No_squeeze)
         v(3)=0.
         write (66,'(3(A,1PE10.3),A)')
     $   'Output: v1 = ',v(1),' v2 = ',v(2),' v3 = ',v(3),' : Case 1'

         write (66,'(/)')
         return
      end if
c
c  Close the gap initially
c
      if (inc.le.No_squeeze.and.nsurf.eq.2) then
         if (type.eq.1) then
            v(1)=-delta*float(inc)
     $            /float((No_squeeze*(No_squeeze+1))/2)
            v(2)=0.
            v(3)=0.
         else if (type.eq.2) then
            v(1)=0.
            v(2)=delta*float(inc)
     $            /float((No_squeeze*(No_squeeze+1))/2)
            v(3)=0.
         else
            v(1)=0.
            v(2)=-delta*float(inc)
     $            /float((No_squeeze*(No_squeeze+1))/2)
            v(3)=0.
         end if
         write (66,'(3(A,1PE10.3),A)')
     $   'Output: v1 = ',v(1),' v2 = ',v(2),' v3 = ',v(3),' : Case 2'
         write (66,'(/)')
         return
      end if
```

```
C
C  Find the velocities during cyclic loading
C
      if (inc.gt.20.and.nsurf.eq.2) then
         if (type.eq.1) then
            v(1)=-d_delta*omega*cos(omega*(time
     $          -float(No_squeeze+No_press)))
            v(2)=0.
            v(3)=0.
         else if (type.eq.2) then
            v(1)=0.
            v(2)=d_delta*omega*cos(omega*(time
     $          -float(No_squeeze+No_press)))
            v(3)=0.
         else
            v(1)=0.
            v(2)=-d_delta*omega*cos(omega*(time
     $          -float(No_squeeze+No_press)))
            v(3)=0.
         end if
         write (66,'(3(A,1PE10.3),A)')
     $  'Output: v1 = ',v(1),' v2 = ',v(2),' v3 = ',v(3),' : Case 3'
         write (66,'(/)')
         return
      end if
C
C  Return zero velocity components
C
      write (66,'(3(A,1PE10.3),A)')
     $  'Output: v1 = ',v(1),' v2 = ',v(2),' v3 = ',v(3),' : Case 0'
      write (66,'(/)')
      return
      end
      program make_model
C
C  Make a MARC input deck for an oring finite element model
C
      implicit none
C
C  Declare variables
C
      character*80 line                ! Line of characters
      character*40 title               ! Title
      character*24 jobid               ! Job ID
      integer EOF                      ! End of file flag
      integer type                     ! Element type
      integer oring                    ! O-ring type
      integer m(4,1500)                ! Connectivity
      integer i1,i2,i3,i4              ! Temporary connectivity
      integer No_elems                 ! Number of elements
      integer No_nodes                 ! Number of nodes
      integer n                        ! Counter
      integer length                   ! Length of the jobid string
      integer no_cyc                   ! Number of cycles
      integer No_squeeze               ! Number of squeeze increments
      integer No_press                 ! Number of pressure increments
      integer No_inc_cycle             ! Number of increments per cycle
      real z(1500),r(1500)            ! Coordinates
      real inside_dia                 ! O-ring inside diameter
      real Maj_dia                    ! Major diameter of o-ring
      real Min_dia                    ! Minor diameter of o-ring
      real a                          ! Major radius of o-ring
      real b                          ! Minor radius of o-ring
      real mu                         ! Friction coefficient
      real width,depth                ! O-ring gland dimensions
      real gap,in_dia_edg             ! O-ring gap & inner edge location
      real inner_edg                  ! Inner diameter for inner type 3
      real pint,pext                  ! Internal and external pressure
      real max_squeeze                ! Mean compression squeeze
      real mean_squeeze               ! Max. compression squeeze
      real delta                      ! Compression displacement
      real d_delta                    ! Variation of compression displ.
      real dp                         ! Pressure change
      real phi                        ! Pressure phase angle
      real freq                       ! Frequency (Hz)
      real omega                      ! Frequency (rad/sec)
      real z1,r1                      ! Temporary coordinates
      real pi                         ! 3.14159265358979
      real Gp,tan_delta,freq_max      ! Shear modulus,tan(delta)
                                      ! & freq. w. max tan(delta)
c
c  Set constants
c
      pi=3.14159265358979
      jobid='
c
c  Get the job name
c
      EOF=0
      open (unit=10,file='jobname.txt',status='old',iostat=EOF)
      if (EOF.eq.0) then
         read (10,'(A)') jobid
      else
         write (6,'("Please enter jobid: ",$)')
         read (5,'(A)') jobid
      end if
      close (unit=10)
c
c  Get the input data from the user and echo to screen
c
      open (unit=6,carriagecontrol='FORTRAN')
      write (6,'(A,$)') 'Please enter a title for the case: '
      read (5,'(A)') title
      write (6,'(A,$)')
     $   'Please enter the inside, cross-section diameter: '
      read (5,*) Inside_dia,Min_dia
      write (6,'(A)')
     $   'Please enter at frequency with maximum tan(delta): '
      write (6,'(A,$)') '    (1) G-prime: '
      read (5,*) Gp
      write (6,'(A,$)') '    (2) Tan(delta): '
      read (5,*) tan_delta
      write (6,'(A,$)') '    (3) Frequency: '
      read (5,*) freq_max
       write (6,'(A,$)')
     $                  'Please enter friction coeffifient: '
      read (5,*) mu
      write (6,'(A)') 'O-ring type     Description'
      write (6,'(A)') '     1             Face'
      write (6,'(A)') '     2             Rod'
      write (6,'(A)') '     3            Piston'
      write (6,'(A,$)')
     $                  'Please enter the oring type: '
      oring=0
      do while (oring.lt.1.or.oring.gt.3)
         read (5,*) oring
         if (oring.lt.1.or.oring.gt.3) then
            write (6,'(A,$)')
     $                  'Type must be 1, 2, or 3'
            write (6,'(A,$)')
     $                  'Please enter the oring type: '
         end if
      end do
      write (6,'(A,$)')
     $                  'Please enter gland width & depth: '
      read (5,*) width,depth
      write (6,'(A,$)')
     $                  'Please enter the inner edge diameter & gap: '
      read (5,*) in_dia_edg,gap
      write (6,'(A,$)')
     $                  'Please enter internal & external pressure: '
      read (5,*) pint,pext
      write (6,'(A,$)')
     $                  'Please enter the mean compression squeeze '
      write (6,'(A,$)')
     $                  'as a fraction of the minor diameter:'
      read (5,*) mean_squeeze
      write (6,'(A,$)')
     $                  'Please enter the max. compression squeeze '
      write (6,'(A,$)')
     $                  'as a fraction of the minor diameter:'
      read (5,*) max_squeeze
      write (6,'(A,$)') 'Please enter the internal pressure change '
      write (6,'(A,$)')  ,'& phase (deg): '
      read (5,*) dp,phi
      write (6,'(A,$)') 'Please enter the number of cycles '
     $                  ,'& frequency: '
      read (5,*) no_cyc,freq
      write (6,'(A)')
     $   'Please enter the number of squeeze, pressure increments'
      write (6,'(A,$)') ' & increments per cycle'
      read (5,*) No_squeeze,No_press,No_inc_cycle
C
C  Find related parameters
C
```

```
      Maj_dia=inside_dia+Min_dia
      inner_edg=0.
      delta=mean_squeeze*Min_dia
      d_delta=(max_squeeze-mean_squeeze)*Min_dia
      phi=pi*phi/180.
      omega=2.*pi*freq
      if (gap.le.0.) gap=Min_dia-depth
      PRINT*,'gap after if (gap.le.0.) gap=Min_dia-depth: ',gap
      if (in_dia_edg.le.0.) then
         if (oring.eq.1) in_dia_edg=0.50*(Maj_dia-Min_dia)
      end if
      if (oring.eq.2) in_dia_edg=-Min_dia/2.
      if (oring.eq.3) then
         if (in_dia_edg.gt.0) then
            inner_edg=in_dia_edg
            in_dia_edg=-Min_dia/2.
            delta=delta-(inner_edg-inside_dia)/2.
         else
            inner_edg=Maj_dia-Min_dia
            in_dia_edg=-Min_dia/2.
         end if
      end if
C
C  Find the first blank of jobid
C
      length=0
      do while (jobid(length:length).ne.' ')
         length=length+1
      end do
      length=length-1
C
C  Echo the input data to the screen
C
      write (6,'(//)')
      write (6,'(1H ,A,A40 /)') 'Title:  ',Title
      write (6,'(1H ,2(A,1PE10.3))')
     $                           'Inside diameter   = ',Maj_dia
     $                       ,'   Xsect diameter    = ',Min_dia
      write (6,'(1H ,A,1PE10.3)')
     $                           'Storage modulus   = ',Gp
      write (6,'(1H ,2(A,1PE10.3),A)')
     $                       ,   'Where Tan(delta)  = ',tan_delta
     $                       ,'   is Max. at',freq_max,' Hz'
      write (6,'(1H ,A,1PE10.3)')
     $                           'Coef. of Friction = ',mu
      if (oring.eq.1) write (6,'(1H ,A)') 'O-ring type: 1 - Face'
      if (oring.eq.2) write (6,'(1H ,A)') 'O-ring type: 2 - Rod'
      if (oring.eq.3) write (6,'(1H ,A)') 'O-ring type: 3 - Piston'
      write (6,'(1H ,2(A,1PE10.3))')
     $                           'Gland width       = ',width
     $                       ,'   Gland depth       = ',depth
      write (6,'(1H ,2(A,1PE10.3))')
     $                           'Inner edge site   = ',in_dia_edg
     $                       ,'   Gap               = ',gap
      if (oring.eq.3) write (6,'(1H ,A,1PE10.3)')
     $                            'Inner diameter    = ',inner_edg
      write (6,'(1H ,2(A,1PE10.3))')
     $                            'Internal pressure = ',pint
     $                        ,'   External pressure = ',pext
      write (6,'(1H ,2(A,1PE10.3))')
     $                            'Mean squeeze (%)  = ',mean_squeeze
     $                        ,'   Max. squeeze (%)  = ',max_squeeze
      write (6,'(1H ,2(A,1PE10.3),A)')
     $                            'Pressure change   = ',dp
     $                        ,'   & Phase angle     = ',phi*180./pi,' Hz'
      write (6,'(1H ,A,I5,A,1PE10.3)')
     $                            'Number of cycles  = ',no_cyc
     $                  ,'         Drive Frequency   = ',freq
C
C  Send the summary of the input data to a file
C
      open (unit=3,file=jobid(1:length)//'.sum',status='unknown')
      write (3,'(//)')
      write (3,'(1H ,A,A40 /)') 'Title:  ',Title
      write (3,'(1H ,2(A,1PE10.3))')
     $                            'Inside diameter   = ',Maj_dia
     $                        ,'   Xsect diameter    = ',Min_dia
      write (3,'(1H ,A,1PE10.3)')
     $                            'Storage modulus   = ',Gp
      write (3,'(1H ,2(A,1PE10.3),A)')
     $                        ,   'Where Tan(delta)  = ',tan_delta
     $                        ,'   is Max. at',freq_max,' Hz'
      write (3,'(1H ,A,1PE10.3)')
     $                            'Coef. of Friction = ',mu
      if (oring.eq.1) write (3,'(1H ,A)') 'O-ring type: 1 - Face'
      if (oring.eq.2) write (3,'(1H ,A)') 'O-ring type: 2 - Rod'
      if (oring.eq.3) write (3,'(1H ,A)') 'O-ring type: 3 - Piston'
      write (3,'(1H ,2(A,1PE10.3))')
     $                            'Gland width       = ',width
     $                        ,'   Gland depth       = ',depth
      write (3,'(1H ,2(A,1PE10.3))')
     $                            'Inner edge site   = ',in_dia_edg
     $                        ,'   Gap               = ',gap
      if (oring.eq.3) write (3,'(1H ,A,1PE10.3)')
     $                           'Inner diameter    = ',inner_edg
      write (3,'(1H ,2(A,1PE10.3))')
     $                            'Internal pressure = ',pint
     $                        ,'   External pressure = ',pext
      write (3,'(1H ,2(A,1PE10.3))')
     $                            'Mean squeeze (%)  = ',mean_squeeze
     $                        ,'   Max. squeeze (%)  = ',max_squeeze
      write (3,'(1H ,2(A,1PE10.3),A)')
     $                            'Pressure change   = ',dp
     $                        ,'   & Phase angle     = ',phi*180./pi,' Hz'
      write (3,'(1H ,A,I5,A,1PE10.3)')
     $                            'Number of cycles  = ',no_cyc
     $                  ,'         Drive Frequency   = ',freq
      close (unit=3)
```

```
C
C  Read the normalized nodal point quantities
C
      open (unit=10,file='coor.inp',status='old')
      EOF=0
      No_nodes=0
      do while(EOF.eq.0)
         read (10,'(I5,2F10.0)',iostat=EOF) n,z1,r1
         if (EOF.eq.0) then
            z(n)=z1
            r(n)=r1
            No_nodes=No_nodes+1
         end if
      end do
      No_nodes=No_nodes+1
      z(No_nodes)=0.
      r(No_nodes)=0.
      close(unit=10)
C
C  Read the connectivity
C
      open (unit=10,file='conn.inp',status='old')
      EOF=0
      No_elems=0
      do while(EOF.eq.0)
         read (10,'(6I5)',iostat=EOF) n,type,i1,i2,i3,i4
         if (EOF.eq.0) then
            m(1,n)=i1
            m(2,n)=i2
            m(3,n)=i3
            m(4,n)=i4
            No_elems=No_elems+1
         end if
      end do
C
C  Open the MARC data file & write the title card
C
      open (unit=20,file=jobid(1:length)//'.dat',status='unknown')
      write(20,'(A,A)') 'title     ',Title
C
C  Read the MARC heading cards & write to MARC data file
C
      open (unit=10,file='head.inp',status='old')
      EOF=0
      do while(EOF.eq.0)
         read (10,'(A80)',iostat=EOF) line
         if (EOF.eq.0) write (20,'(A)') line
      end do
      close (unit=10)
C
C  Read the set definition cards & write to MARC data file
C
      open (unit=10,file='presure_sets.inp',status='old')
      EOF=0
      do while(EOF.eq.0)
         read (10,'(A80)',iostat=EOF) line
         if (EOF.eq.0) write (20,'(A)') line
      end do
      close (unit=10)
C
C  Write the connectivity
C
      type=82
      write (20,'(A)') 'connectivity'
      write (20,'(A)') ' '
      do n=1,No_elems
         write (20,'(7I5)')
     $      n,type,m(1,n),m(2,n),m(3,n),m(4,n),No_nodes+n
      end do
C
C  Write the nodes (rotate normalized grid 45 degrees, translate and scale)
C
      a=Maj_dia/2.
      b=Min_dia/2.
      write (20,'(A)') 'coordinates'
      write (20,'(A,I5)') '    3',No_nodes
      do n=1,No_nodes
         if (oring.eq.1) write (20,'(I5,1P2E10.3)')
     $      n,b*(r(n)+z(n))/sqrt(2.),b*(r(n)-z(n))/sqrt(2.)+a
         if (oring.eq.2) write (20,'(I5,1P2E10.3)')
     $      n,b*(-r(n)+z(n))/sqrt(2.),b*(r(n)+z(n))/sqrt(2.)+a
         if (oring.eq.3) write (20,'(I5,1P2E10.3)')
     $      n,b*(-r(n)+z(n))/sqrt(2.),b*(r(n)+z(n))/sqrt(2.)+a
      end do
C
C  Write the spring block & fix the displacement of the added node
C
      write (20,'(A)') 'springs'
      write (20,'(A,I5,A)') '633,1,',No_nodes,',1,1.e-1'
      write (20,'(A,I5,A)') '633,2,',No_nodes,',2,1.e-1'
      write (20,'(A)') 'fixed disp'
      write (20,'(A)') ' '
      write (20,'(A)') '0.,0.'
      write (20,'(A)') '1,2'
      write (20,'(I5)') No_nodes
C
C  Write the material properties
C
      write (20,'(A)') 'mooney'
      write (20,'(A)') '1,'
      write (20,'(A)') '1,'
      write (20,'(1PE10.3)') Gp*(1.+tan_delta)
      write (20,'(A,I5)') '1 TO',No_elems
      write (20,'(A)') 'viscelmoon'
      write (20,'(A)') '1,'
      write (20,'(A)') '1,1'
      write (20,'(1P2E10.3)') 2.*tan_delta/(1.+tan_delta),1./omega
C
C  Write the contact data
C
```

```
 write (20,'(A)') 'contact'
 write (20,'(A)') '3,5000,1500,2,1,0,0'
 write (20,'(A)') ', , ,.005'
 write (20,'(A)') '1,0,,,'
 write (20,'(A,1PE10.3)') ',,,,,,',mu
 write (20,'(A,I5)') '1 TO',No_elems
 write (20,'(A)') '2,1'
 if (oring.eq.1) then
    write (20,'(1PE10.3,A,1PE10.3)') 1.0001*b,',,,-0.010,,,',mu
 else if (oring.eq.2) then
    write (20,'(1PE10.3,A,1PE10.3)') 1.0001*b,',,,,0.010,,',mu
 else
    write (20,'(1PE10.3,A,1PE10.3)') 1.0001*b,',,,,-0.010,,',mu
 end if
 write (20,'(A)') '1,2'
 if (oring.eq.1) then
    write (20,'(1PE10.3,A,1PE10.3)')
$   -b+depth+gap,',',in_dia_edg+width+b
    write (20,'(1PE10.3,A,1PE10.3)')
$   -b+depth+gap,',',in_dia_edg-b
 else if (oring.eq.2) then
    write (20,'(1PE10.3,A,1PE10.3)')
$   b+width/2.,',',a+b-depth-gap
    write (20,'(1PE10.3,A,1PE10.3)')
$   -b-width/2.,',',a+b-depth-gap
 else
     write(20,'(1PE10.3,A,1PE10.3)')
$   -b-width/2.,',',a-b+depth+gap
    write (20,'(1PE10.3,A,1PE10.3)')
$   b+width/2.,',',a-b+depth+gap
 end if
 write (20,'(A)') '3,1'
 write (20,'(1PE10.3,A,1PE10.3)') -1.0001*b,',,,,,,',mu
 if (depth.gt.0.) then
    write (20,'(A)') '1,6'
 else
    write (20,'(A)') '1,2'
 end if
 if (oring.eq.1) then
    write (20,'(1PE10.3,A,1PE10.3)')
$   -b+depth,',',in_dia_edg-b
    if (depth.gt.0.) then
       write (20,'(1PE10.3,A,1PE10.3)')
$      -b+depth,',',in_dia_edg
       write (20,'(1PE10.3,A,1PE10.3)')
$      -b,',',in_dia_edg
       write (20,'(1PE10.3,A,1PE10.3)')
$      -b,',',in_dia_edg+width
       write (20,'(1PE10.3,A,1PE10.3)')
$      -b+depth,',',in_dia_edg+width
    end if
    write (20,'(1PE10.3,A,1PE10.3)')
$   -b+depth,',',in_dia_edg+width+b
 else if (oring.eq.2) then
    write (20,'(1PE10.3,A,1PE10.3)')
$   -b-width/2.,',',a+b-depth
    if (depth.gt.0.) then
```

```
            write (20,'(1PE10.3,A,1PE10.3)')
     $      - Min_dia/2.,',',a+b-depth
            write (20,'(1PE10.3,A,1PE10.3)')
     $      - Min_dia/2.,',',a+b
            write (20,'(1PE10.3,A,1PE10.3)')
     $      -Min_dia/2.+width,',',a+b
            write (20,'(1PE10.3,A,1PE10.3)')
     $      -Min_dia/2.+width,',',a+b-depth
         end if
         write (20,'(1PE10.3,A,1PE10.3)')
     $   b+width/2.,',',a+b-depth
      else
         write (20,'(1PE10.3,A,1PE10.3)')
     $   b+width/2.,',',a-b+depth
         if (depth.gt.0.) then
            write (20,'(1PE10.3,A,1PE10.3)')
     $      width-Min_dia/2.,',',a-b+depth
            write (20,'(1PE10.3,A,1PE10.3)')
     $      width-Min_dia/2.,',',a-b
            write (20,'(1PE10.3,A,1PE10.3)')
     $      -Min_dia/2.,',',a-b
            write (20,'(1PE10.3,A,1PE10.3)')
     $      -Min_dia/2.,',',a-b+depth
         end if
         write (20,'(1PE10.3,A,1PE10.3)')
     $   -b-width/2.,',',a-b+depth
      end if
      write (20,'(A)') 'contact table'
      write (20,'(A)') '1,'
      write (20,'(A,2(1PE10.3,A),1PE10.3)')
     $      '1,',1.E-2*b,',',1.e-1*2.*pi**2*a*b*Gp*gap,',',mu
      write (20,'(A)') '2,3'
C
C  Read the MARC data to end the model definition  & write to MARC data file
C
      open (unit=10,file='endopt.inp',status='old')
      EOF=0
      do while(EOF.eq.0)
         read (10,'(A80)',iostat=EOF) line
         if (EOF.eq.0) write (20,'(A)') line
      end do
      close (unit=10)
C
C  Write the control block
C
      open (unit=10,file='control.inp',status='old')
      EOF=0
      do while(EOF.eq.0)
         read (10,'(A80)',iostat=EOF) line
         if (EOF.eq.0) write (20,'(A)') line
      end do
      close (unit=10)
C
C  Write the rigid body surface motion squeeze increments
C
```

```
      write (20,'(A)') 'motion change'
      write (20,'(A)') '2,'
      write (20,'(A)') '2,'
      write (20,'(3(E10.3,A),E10.3)')
     $   0.,',',0.,',',0.,',',mu
      write (20,'(A)') '3,'
      write (20,'(3(E10.3,A),E10.3)')
     $   0.,',',0.,',',0.,',',mu
      write (20,'(A)') 'time step'
      write (20,'(A)') '1.,'
      write (20,'(A)') 'auto load'
      write (20,'(I5)') No_squeeze
      write (20,'(A)') 'continue'
C
C  Hold the rigid bodies still
C
      write (20,'(A)') 'motion change'
      write (20,'(A)') '1,'
      write (20,'(A)') '2,'
      write (20,'(3(E10.3,A),E10.3)')
     $   0.,',',0.,',',0.,',',mu
C
C  Write the pressure data file
C
      open (unit=10,file='dist.inp',status='old')
      EOF=0
                                     ! Count the lines
      n=0
      do while(EOF.eq.0)
         read (10,'(A80)',iostat=EOF) line
         if (EOF.eq.0)  then
            n=n+1
            if (n.eq.3.or.n.eq.5.or.n.eq.7) then
               if (n.eq.3)
     $            write (20,'(A,A)') '    3','1.'
               if (n.eq.5)
     $            write (20,'(A,A)') '    9','1.'
               if (n.eq.7)
     $            write (20,'(A,A)') '   11','1.'
            else if (n.eq.12.or.n.eq.14.or.n.eq.19) then
               if (n.eq.12)
     $            write (20,'(A,A)') '    3','1.'
               if (n.eq.14)
     $            write (20,'(A,A)') '    7','1.'
               if (n.eq.19)
     $            write (20,'(A,A)') '    9','1.'
            else
               write (20,'(A)') line
            end if
         end if
      end do
      close (unit=10)
      write (20,'(A)') 'time step'
      write (20,'(A)') '1.,'
      write (20,'(A)') 'auto load'
      write (20,'(I5)') No_press
      write (20,'(A)') 'continue'
```

```
C
C  Write the rigid body surface motion harmonic increments
C
      write (20,'(A)') 'motion change'
      write (20,'(A)') '2,'
      write (20,'(A)') '2,'
      write (20,'(3(E10.3,A),E10.3)')
     $   0.,',',0.,',',0.,',',mu
C
C  Write the pressure data file
C
      open (unit=10,file='dist.inp',status='old')
      EOF=0
      n=0                               ! Count the lines
      do while(EOF.eq.0)
         read (10,'(A80)',iostat=EOF) line
         if (EOF.eq.0)  then
            n=n+1
            if (n.eq.3.or.n.eq.5.or.n.eq.7) then
               if (n.eq.3)
     $            write (20,'(2A)') '    3',' 1.'
               if (n.eq.5)
     $            write (20,'(2A)') '    9',' 1.'
               if (n.eq.7)
     $            write (20,'(2A)') '   11',' 1.'
            else if (n.eq.12.or.n.eq.14.or.n.eq.19) then
               if (n.eq.12)
     $            write (20,'(2A)') '    3',' 1.'
               if (n.eq.14)
     $            write (20,'(2A)') '    7',' 1.'
               if (n.eq.19)
     $            write (20,'(2A)') '    9',' 1.'
            else
               write (20,'(A)') line
            end if
         end if
      end do
      close (unit=10)
      write (20,'(A)') 'time step'
      write (20,'(1PE10.3,A)') 1./(float(No_inc_cycle)*freq),','
      write (20,'(A)') 'auto load'
      write (20,'(I5)') No_inc_cycle*no_cyc
      write (20,'(A)') 'continue'
C
C  Close the MARC data file
C
      close(unit=20)
C
C  Write out the pressure and gap parameters to a file
C
```

```
      open (unit=21,file='temp.inp',status='unknown')
      write (21,*) delta
      write (21,*) d_delta
      write (21,*) pint
      write (21,*) pext
      write (21,*) dp
      write (21,*) omega
      write (21,*) phi
      write (21,*) oring
      write (21,*) inner_edg-inside_dia
      write (21,*) No_squeeze
      write (21,*) No_press
      close(unit=21)
C
C  Normal end
C
      stop
      end
```

APPENDIX D
ANSYS MODELING USER'S MANUAL

The O-ring ANSYS version cannot simulate the diverse problems that the O-ring MARC version can, but the ANSYS version can readily model the parallel plate cyclic tests. To perform these simulations, an ANSYS 5.4 macro (see the listing at the end of this appendix) was written which prompts the user for geometric and material data, generates an axisymmetric model of a viscoelastic rubber o-ring, and runs a compression and decompression cycle using rigid compression surfaces.

Directions:

1. Enter ANSYS
2. Type 'oring' at the command line (executes the macro 'oring.mac')
3. Answer the following prompts:
   - Job Name (This will be the name of the database file containing the model)
   - Title (This will be listed at the bottom of all of the output plots)
   - Major Diameter (see Figure D.1 below)
   - Minor Diameter (see Figure D.1 below)
   - Compression Displacement (Total compression of the O-ring (see Figure D.1))
   - Compression Velocity (Velocity of the rigid surfaces into the O-ring during compression only)
   - Instantaneous shear modulus (G(0) in the viscoelastic constitutive model)
   - Shear modulus at infinite time (G(inf.) in the viscoelastic constitutive model)
   - Relaxation time factor (λ in the viscoelastic constitutive model)

Model Execution:

Once the user completes the entry of geometry and material variables, the macro generates the geometry and the mesh, defines the rigid "target" surfaces, and executes three load steps. Several material parameters are currently set by the macro (although they can readily be modified by editing the macro). These parameters are as follows:

$G = G(\text{inf.})$ (The global shear modulus is set equal to G(inf.) in the viscoelastic model
$\nu = 0.475$ (Poisson's ratio)
$\mu = 0.0$ (Friction coefficient - Zero friction assumed)

The ANSYS automesher is invoked to generate the mesh in the O-ring. It currently uses 60 higher-order viscoelastic elements around the periphery of the circle defining the cross-section. The axisymmetric option is turned on to fully account for the true stress state in the O-ring. The large deformation option is also invoked to account for large strain and large displacement effects.

The load steps are set up to run the following sequence of compression, hold, and decompression:

Load Steps

1. Compression:	time = Δ/v	(in 10 - 30 substeps)
	Δ = compression displacement	
	v = compression velocity	
Hold:	time = $5*\lambda$	(in 10 substeps)
Decompression:	time = $1*\lambda$	(in 10 - 30 substeps)

The solver is automatically executed for each load step and the basic stress and displacement output is written to the results file '*.rst' for every substep. During compression, the rigid surfaces are compressed from both sides at a rate of 'v'/2. This is equivalent to a compression velocity of 'v' in the case where only one of the surfaces moves.

ANSYS returns a warning during the simulation indicating that the viscoelastic elements have not been tested with the large strain option. The test cases completed indicate this should not be a problem.

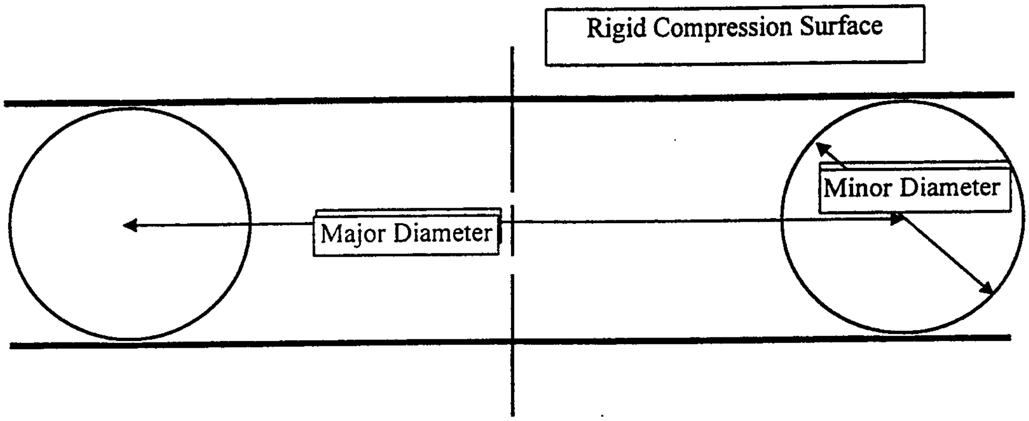


Figure D.1 Geometry of the O-Ring Model

```
!
! ANSYS Macro File - Creates a model of an oring with flat contact surfaces
!                    on each side.  The oring is compressed by one or both of
!                    these surfaces until it reaches a user-defined displacement.
!
! Written by:  Steve Hale (United Technologies Research Center:  (860)610-7910)
!
! Original Date:  2/25/98
! Latest Update:  3/13/98
!
! This version is set up with 8-noded Visco-Elastic elements.
!*****************************************************************************
!
! Obtain parameters
!
pi = acos(-1)
*ask,jobid,'Job Name (default = oring)','oring'
*ask,titl,'Title (default = jobid)',jobid
/title,%titl%
*ASK,maj_dia,Major Diameter (default = 2.0),2.0
*ASK,min_dia,Minor Diameter (default = 0.5),0.5
!*ASK,G,Shear Modulus (default = 1000.),1000.
!*ASK,nu,Poisson Ratio (default = 0.475),0.475
*ASK,delta,Compression Displacement (default = 0.10),0.10
*ASK,vel,Compression Velocity (default = 0.50),0.50
*ASK,G0,Instantaneous Shear Modulus (default = 2300.),2300.
*ASK,G1,Shear Modulus at Infinite Time (default = 1700.),1700.
*ASK,lambda, Relaxation Time Factor (lambda) (default = 0.1),0.1
!*ASK,thold,Hold Time (default = 1 sec.),1.0
!
G = G1       ! Equate the shear modulus to the modulus at time = infinity
!
! Assume an "incompressible" material
!
nu = 0.475
!
! Friction coefficient
!
frict = 0.0
!
! Number of elements around the periphery of the circular area
!
nels = 60.
nell = 60./4
!
! Major and minor radii
!
a = maj_dia/2
b = min_dia/2

! Write out user-input parameters

*MSG,UI,titl,maj_dia,min_dia,G,nu,delta,vel
Job Title = %C %/Major Diam. = %G,  Minor Diam. = %G %/Shear Mod. = %G,  Poisson Ratio =
G %/Comp. Disp. = %G,  Comp. Vel. = %G
```

```
! Set up a structural analysis run

KEYW,PR_SET,1
KEYW,PR_STRUC,1

! Preprocessor

/PREP7
!*
!** Old **
!ET,1,PLANE42
!
!** New: 8-noded visco-elastic element **
ET,1,VISCO88
!*
KEYOPT,1,1,0
KEYOPT,1,2,0
KEYOPT,1,3,1
KEYOPT,1,5,0
KEYOPT,1,6,0
KEYOPT,1,7,0
!*
Emod = 2.*(1.+nu)*G
!*

! Enter material properties

MP,EX,1,Emod,    !** Young's modulus
MP,NUXY,1,nu,    !** Poisson's ratio
MP,MU,1,frict,   !** Friction coefficient

!
! Set up the VISCOELASTIC constants
!
!lambda = 0.10              !** Relaxation time factor
!G0 = 2300.                 !** Shear modulus at time= 0
!G1 = 1700.                 !** Shear modulus at time= infinity
akval0 = Emod/(3*(0.01))   !** Bulk modulus at time= 0
akval1 = Emod/(3*(0.01))   !** Bulk modulus at time= infinity
!*
TB,EVISC,1, , , ,
!*
TBMODIF,1,1,600
TBMODIF,1,2,1
TBMODIF,1,3,1
TBMODIF,2,1,1
TBMODIF,6,1,0.0001
TBMODIF,7,1,0.0001
TBMODIF,8,1,600
TBMODIF,10,1,G0
TBMODIF,10,2,G1
TBMODIF,10,3,akval0
TBMODIF,10,4,akval1
TBMODIF,10,5,1
TBMODIF,11,1,1
```

```
TBMODIF,13,1,lambda
TBMODIF,15,1,1
TBMODIF,16,1,1
TBMODIF,18,1,0.001
!**
!** Define the reference (zero stress) temp. and the uniform initial temp.
!**
TREF,600.,
TUNIF,600.,

! Set up the Mooney-Rivlin constants
! **Note:  I could not get this model to converge with the Mooney-Rivlin
!          material. (2/25/98)
!
!a10 = G/2.
!a01 = 0.0
!TB,MOONEY,1
!TBDATA,1,a10
!TBDATA,2,a01

! Create a circular area
CYL4,a,0.,b

! Set up a mesh size and run the automated mesher

LESIZE,ALL, , ,nell,1,1
MSHKEY,0
CM,_Y,AREA
ASEL, , , ,       1
CM,_Y1,AREA
CHKMSH,'AREA'
CMSEL,S,_Y
!*
AMESH,_Y1
!*
CMDEL,_Y
CMDEL,_Y1
CMDEL,_Y2
!*
!**********************************************
! Create the rigid target CONTACT surfaces
!
ET,2,TARGE169
TSHAPE,LINE
!*
K,5,0,b+.01,0,
K,6,a+2*b,b+.01,0,
K,7,a+2*b,-b-.01,0,
K,8,0,-b-.01,0,
!*
LSTR,       5,       6
LSTR,       7,       8
!*
TYPE,   2
MAT,       1
REAL,       2
```

```
!*
LMESH,        5
KMESH,        5
!*
REAL,        3
!*
LMESH,        6
KMESH,        7
!*
!
! Create the flexible CONTACT surfaces
! (contact for a quadratic element)
!
ET,3,CONTA172
!*
!*
TYPE,    3
MAT,        1
REAL,        2
!*
LSEL,S,LOC,X,a-b,a+b
LSEL,R,LOC,Y,-b,b
NSLL,S,1
NSEL,R,LOC,Y,0.,b
!*
!ESURF,_Z1,TOP
ESURF,,TOP
!*
MAT,        1
REAL,        3
!*
NSLL,S,1
NSEL,R,LOC,Y,0.,-b
!*
!ESURF,_Z1,TOP
ESURF,,TOP
!*
ALLSEL
!*********************************************
!
! Generate real constants for the CONTACT elements
!
R,2,0,0,.5,.05,0.02,2.,
RMORE,0.02,0,0,
!*
R,3,0,0,.5,0.05,0.02,2.,
RMORE,0.02,0,0,
!*
NLGEOM,1
LUMPM,0
EQSLV,FRONT,1e-08,0,
SSTIF,OFF
PSTRES,OFF
!
KEYOPT,3,7,1
!
! Fix the Oring from rigid body motion in the axial direction
!
```

```
NSEL,ALL
DK,3,UY,0.0,
DK,3,ROTZ,0.0,
DK,1,UY,0.0,
!
DK,5,UX,0.0,
DK,8,UX,0.0,
DK,5,ROTZ,0.0,
DK,8,ROTZ,0.0,
DK,6,UX,0.0,
DK,7,UX,0.0,
DK,6,ROTZ,0.0,
DK,7,ROTZ,0.0,
DTRAN
!
!********************
!
! Define the solution and load step options
!
/SOLU
NLGEOM,ON
NROPT,AUTO, ,OFF
EQSLV,FRONT,1e-08,,
!*
NEQIT,25
LNSRCH,ON
!*
PRED,ON,,ON
!*
!SOLCON,ON,1
AUTOTS,0
NEQIT,25
KBC,0
!*
!* Set the force convergence tolerance
!CNVTOL,M,,,,0.9,
CNVTOL,F,,,,0.25
!********************
!
! Apply the compressive displacement to the nodes of the rigid surface
! and fix the x and z-rotation degrees of freedom of these nodes.
!
! ** Step 1 **
!
v2 = vel/2.
d2 = delta/2.
!
NSEL,S,LOC,Y,b+.001,b+.05
d,all,uy,-d2
d,all,ux,0.
d,all,rotz,0.
NSEL,ALL
!
NSEL,S,LOC,Y,-b-.05,-b-.001
d,all,uy,d2
d,all,ux,0.
```

```
d,all,rotz,0.
ALLSEL
!*
time1 = delta/vel
TIME,time1
!
!* Set the number of substeps
!
ratio = delta/min_dia
*if,ratio,LE,0.10,then
  istep1 = 10
  npr = 1
*elseif,ratio,LE,0.20,then
  istep1 = 20
  npr = 2
*else
  istep1 = 30
  npr = 3
*endif
NSUBST,istep1
!*
!* Write all output at every <npr> substep
!OUTRES,ALL,npr
!* Write only the basic quantities at all substeps
OUTRES,BASIC,1
!
!LSWRITE,1,
SOLVE
!*******************
!
! Hold at their current positions for 5 times the relaxation time factor
!
! ** Step 2 **
!
time2 = 5*lambda + time1
!
!* Set the hold time and the number of substeps
!
TIME,time2
istep2 = 10
NSUBST,istep2
!*
!* Write all output every substep
OUTRES,BASIC,1
!
!LSWRITE,2,
SOLVE
!*******************
```

```
!
! Move the surfaces back to their original positions (y-disp. of zero)
!  rapidly (in 'lambda' seconds)
!
! ** Step 3 **
!
time3 = lambda + time2
v2 = 0.0
d2 = 0.0
!
NSEL,S,LOC,Y,b+.001,b+.05
d,all,uy,-d2
NSEL,ALL
!
NSEL,S,LOC,Y,-b-.05,-b-.001
d,all,uy,d2
ALLSEL
!
!* Set the displacement time and the number of substeps
!
TIME,time3
istep3 = istep1
NSUBST,istep3
!*
!* Write all basic output every substep
OUTRES,BASIC,1
!
!LSWRITE,3,
SOLVE
!********************
!
SAVE,jobid,db,
/EOF
```

APPENDIX E
PENDULUM TEST REDUCTION SOFTWARE

The software for interpreting the pendulum test data provides the user with a simple and effective way to obtain estimates for the O-ring material properties. The code was written in Fortran 77 and a listing of the code is included at the end of this appendix. Parameters are set at the top of the main program followed by declarations for each variable in the main program. All variables must be declared if the implicit none statement at the fifth line of the program remains. Declaring all variables allows the programmer, and subsequent users, to describe each variable, and removes many potential errors, making debugging a simpler task. After the declarations some constants the user may want to redefine are set. The program then asks the user for the job name. This is a unique name, that will be appended by extensions for various files associated with the test being analyzed. The program then asks the user for the initial angle, and echoes the job name and the angle to standard output. The program next opens an output file, on unit 12 as a file with the job name appended by .out, and writes the job name and the initial angle.

Next the data for the pendulum characteristics are read from a file named 'pendulum.dat' (an example is shown in Figure E.1) and echoed to standard output. The geometry for the O-ring is stored in a file separately by the user (se, for example, FigureE.2.) Now the user supplies the name of the O-ring data file. The O-ring data consists of the major (inside diameter plus cross-section radius) and minor diameter (cross-section diameter.)

The software is now ready to read the transient angle for the pendulum arm. The data is stored in a file with the job name appended with a .txt or .TXT. If the file is improperly named the program writes a warning message and stops. The first two lines of the data include a title, and column headings. Now the program can read the time and the signal strength, which is proportional to the angle. A count of the number of data points is also completed. The initial data is at the starting angle, allowing the proportionality constant to be found. The signal strength is next converted to angle.

The initial energy is found next by using the initial angle. The program now begins the loop over each bounce after writing the column headings. The loop over each bounce consists of first finding the next minimum in the angle, and then interpolating to find the times when the angle is past zero (i.e., in contact with the pendulum.) The next maximum can now be found form the data. It is now possible to find all the parameters for the O-ring properties using equations (24) through (38) in section 7.3.2. The results are sent to standard output and unit 12 for each bounce.

```
320.                ! mass (g)
28.25               ! k_gy (cm)
33.02               ! Length (cm)
1.016               ! chord (cm)
24.13               ! cg (cm)
980.                ! g (cm/s^2)
```

Figure E.1

```
2.856               ! Major diameter of O-ring
0.3175              ! Minor diameter of O-ring
```

Figure E.2

```
      program interpert
C
C  Reduce the transient pendulum test data
C
      implicit none
C
C  Declare paramaters
C
      integer MAX                      ! Maximum number of data points in a test
      integer TESTS                    ! Maximum number of tests in a set
      parameter (MAX=10001)
      parameter (TESTS=5)
C
C  Declare variables
C
      character*24 job_name            ! Name of job to be run
      character*24 oring_file          ! Name of O-ring data file
      character*80 line                ! Line of data
      character*12 title               ! Case title
      real theta(MAX,TESTS)            ! Measured angular position
      real time(MAX,TESTS)             ! Time at measured angle
      real theta_in                    ! An input value for theta
      real time_in                     ! An input value for time
      real initial_deg                 ! Initial angle in degrees
      real pi                          ! 3.14159265358979
      real volts_rads                  ! Calibration constant
      real volts_zero                  ! Zero angle in volts
      real mass                        ! Pendulum mass
      real k_gy                        ! Pendulum radius of gyration
      real Length                      ! Pendulum length to impact point
      real chord                       ! Pendulum chord length at impact point
      real cg                          ! Distance to c.g.
      real g                           ! Acceleration of gravity
      real minor_diameter              ! O-ring minor diameter
      real major_diameter              ! O-ring major diameter
      real minor_radius                ! O-ring minor radius
      real major_radius                ! O-ring major radius
      integer n(TESTS)                 ! Number of data points
      integer n_test                   ! Number of data sets in file
      integer i_test                   ! Data set test number
      integer n_zero                   ! Index for last zero measure
      integer n_rise                   ! Index for last rise to starting angle
      integer n_start                  ! Index for last initial angle measure
      integer n_stop                   ! Index for last data measure
      real zero                        ! Zero reference voltage
      real start                       ! Starting reference voltage
      real theta_max                   ! Current maximum angle
      real theta_dot_old               ! Old maximum angular velocity
      real delta_old                   ! Old maximum displacemnet in o-ring
      real theta_dot_max               ! Maximum angular velocity (@ impact)
      real Energy                      ! Energy of system between bounces
      real dt                          ! Zero crossing interval
      real zeta                        ! Percent of critical damping
      real tan_delta                   ! G''/G'
      real G_star                      ! Total shear modulus
      real Gp                          ! G'
      real Gpp                        ! G''
      real omega                      ! Natural frequency of O-ring
      real stiff                      ! Nondimensional stiffness function
      real pot                        ! Nondimensional energy function
      real disp                       ! Nondimensional displacement function
      integer last                    ! Last character location of job name
      real k                          ! O-ring spring stiffness
      real c                          ! O-ring damping constant
      real alpha                      ! Force approximation coefficient
      integer EOF                     ! Read/write error flag
      integer i,j,j1                    ! Counters
C
C  Set constants
C
      pi=3.14159265358979
      alpha=3.7
C
C Get job name
C
      write (6,'()')
      write (6,'(" Please enter the job name :",$)')
      read (*,'(A)') job_name
      write (6,'()')
C
C  Get the maximum starting angle
C
      write (6,'()')
      write (6,'(" Please enter the initial angle in degrees:",$)')
      read (*,*) initial_deg
      write (6,'()')
      last=index(job_name,' ')-1
      write (6,'(//)')
      write (6,'(A,F8.3,A)') 'Initial angle = ',initial_deg,' deg'
C
C  Open the disk output file & write initial data
C
      open (unit=12,file=job_name(1:last)//'.out',status='unknown')
      write (12,'(A,A)') 'Case: ',job_name
      write (12,'(A,F8.3,A)') 'Initial angle = ',initial_deg,' deg'
C
C  Read & echo pendulum data
C
      open (unit=10,file='pendulum.dat',status='old')
      read (10,*) mass
      read (10,*) k_gy
      read (10,*) Length
      read (10,*) chord
      read (10,*) cg
      read (10,*) g
      close(unit=10)
      write (6,'(//)')
      write (6,'(A)') 'Pendulum Characteristics'
      write (6,'(A,1PE10.3)') 'mass   = ',mass
      write (6,'(A,1PE10.3)') 'k_gy   = ',k_gy
      write (6,'(A,1PE10.3)') 'Length = ',Length
      write (6,'(A,1PE10.3)') 'chord  = ',chord
      write (6,'(A,1PE10.3)') 'cg     = ',cg
      write (6,'(A,1PE10.3)') 'g      = ',g
      close (unit=10)
      write (12,'(//)')
      write (12,'(A)') 'Pendulum Characteristics'
      write (12,'(A,1PE10.3)') 'mass   = ',mass
      write (12,'(A,1PE10.3)') 'k_gy   = ',k_gy
      write (12,'(A,1PE10.3)') 'Length = ',Length
      write (12,'(A,1PE10.3)') 'chord  = ',chord
      write (12,'(A,1PE10.3)') 'cg     = ',cg
      write (12,'(A,1PE10.3)') 'g      = ',g
C
C  Read & echo O-ring geometry
C
      write (6,'(" Please enter the oring geometry file name: ",$)')
      read (5,'(A)') oring_file
      open (unit=10,file=oring_file,status='old')
      read (10,*) major_diameter
      major_radius=0.5*major_diameter
      read (10,*) minor_diameter
      minor_radius=0.5*minor_diameter
      write (6,'(//)')
      write (6,'(A)') 'O-ring Geometry'
      write (6,'(A,1PE10.3)') 'D      = ',major_diameter
      write (6,'(A,1PE10.3)') 'd      = ',minor_diameter
      close (unit=10)
      write (12,'(//)')
      write (12,'(A)') 'O-ring Geometry'
      write (12,'(A,1PE10.3)') 'D      = ',major_diameter
      write (12,'(A,1PE10.3)') 'd      = ',minor_diameter
C
C  Open the data file
C
      EOF=0
      open (unit=10,file=job_name(1:last)//'.txt'
     $     ,status='old',iostat=EOF)
      if (EOF.ne.0) then
         EOF=0
         open (unit=10,file=job_name(1:last)//'.TXT'
     $        ,status='old',iostat=EOF)
         if (EOF.ne.0) then
              WRITE(6,'(A)')'****************************************'
              WRITE(6,'(A)')'* ERROR: Data set file name does not *'
              WRITE(6,'(A)')'* exist with a txt or TXT extension   *'
              WRITE(6,'(A)')'****************************************'
              stop
         end if
      end if
C
C  Read the first two lines
C
      read(10,'(A)') title
      read(10,'(A)') line
C
C  Get the input measured data points
C
```

```
      n_test=0
      do while (n_test.lt.TESTS.and.EOF.eq.0)
         read (10,*,iostat=EOF) time_in,theta_in
         if (EOF.eq.0) then
            if (time_in.le.0.) then
               n_test=n_test+1
               if (n_test.gt.TESTS) then
                  WRITE(6,'(A)')'*****************************'
                  WRITE(6,'(A)')'* ERROR: Too many data sets *'
                  WRITE(6,'(A)')'*****************************'
                  stop
               end if
               n(n_test)=0
            end if
            n(n_test)=n(n_test)+1
            if (n(n_test).gt.MAX) then
               WRITE(6,'(A)')'*************************************'
               WRITE(6,'(A)')'* ERROR: Too many points in data set *'
               WRITE(6,'(A)')'*************************************'
               stop
            end if
            time(n(n_test),n_test)=time_in
            theta(n(n_test),n_test)=theta_in
         end if
      end do
      close (unit=10)
      DO j1=1,n_test
C
C  Find the zero angle, starting angle, and data ranges
C
      call
     $  find_ranges(theta(1,j1),n(j1),n_zero,n_rise,n_start,n_stop,MAX)
      call find_zero(theta(1,j1),n_zero,n_stop,zero)
      call find_start(theta(1,j1),n_rise,n_start,start)
C
C  Convert voltages to radians
C
      call make_radians(theta(1,j1),initial_deg,zero,start,n(j1))
C
C  Set first maximum angle to initial angle
C
      theta_max=initial_deg*pi/180.
      Energy=mass*g*cg*(1.-cos(theta_max))
      theta_dot_max=sqrt(Energy/(.5*mass*k_gy**2))
C
C  Set up the loop for finding the properties at each bounce
C
      i=n_start
      write (6,'(//)')
      write (6,'(3A)') '   Frequency    zeta'
     $                  ,'        G*         tan delta'
     $                  ,'    Gp          Gpp'
      write (12,'(//)')
      write (12,'(3A)') '   Frequency    zeta'
     $                  ,'        G*         tan delta'
     $                  ,'    Gp          Gpp'
      do while (i.le.n_stop)
```

```
C
C  Get O-ring natural frequency from zero angle crossings
C
         call find_next_min(theta(1,j1),i,n(j1),j)
         i=j
         call interpolate_maximum(time,theta(1,j1),i,theta_max,dt)
         omega=pi/dt
C
C  Get fraction of critical damping from relative maximum velocities
C
         call find_next_max(theta(1,j1),i,n,j)
         theta_dot_old=theta_dot_max
         i=j
         call interpolate_maximum(time(1,j1),theta(1,j1),i,theta_max,dt)
         theta_max=theta(i,j1)
         Energy=mass*g*cg*(1.-cos(theta_max))
         theta_dot_max=sqrt(Energy/(.5*mass*k_gy**2))
         zeta=alog(theta_dot_old/theta_dot_max)/pi
         i=i+1
C
C  Get equivalent stiffness and mass from natural frequency and damping coef.
C
         k=mass*(omega*k_gy/Length)**2
         c=2.*zeta*mass*omega*(k_gy/Length)**2
C
C  Find the material constants
C
         G_star=alpha*k/chord
         tan_delta=omega*c/k
         Gp=G_star/sqrt(1.+tan_delta**2)
         Gpp=G_star*tan_delta/sqrt(1.+tan_delta**2)
         write(6,'(1P6E12.3)')
     $      omega/(2.*pi),zeta,G_star,tan_delta,Gp,Gpp
         write(12,'(1P6E12.3)')
     $      omega/(2.*pi),zeta,G_star,tan_delta,Gp,Gpp
      end do
      END DO
C
C  Normal end
C
      stop
      end


      subroutine find_ranges(theta,n,n_zero,n_rise,n_start,n_stop)
C
C  Find the integers that specify the starting points of the three data ranges
C
      implicit none
C
C  Declare variables
C
      real theta(10001)                ! Measured angular position
      integer n                        ! Number of data points
      integer n_zero                   ! Index for last zero measure
      integer n_rise                   ! Index for last rise to starting angle
      integer n_start                  ! Index for last initial anglemeasure
      integer n_stop                   ! Index for last data measure
      integer i,j                      ! Counters
```

```
C
C  Test data to determine if the data starts at the zero position
C
      IF (theta(1).lt.0.5) THEN       ! Data sets zero at zero position
C
C  Zero angle position occurs until five rising readings in a row occur .
C
      i=2                  ! Position in data file
      j=0                  ! Count of number of rises in a row
      do while (j.le.5.and.i.le.n)
         if (theta(i).gt.theta(i-1)) then
            j=j+1
         else
            j=0
         end if
         i=i+1
      end do
      if (i.lt.n) then
         n_zero=i-7
      else
         WRITE(6,'(A)') '**********************************************'
         WRITE(6,'(A)') '* ERROR: Data contains only zero calibration *'
         WRITE(6,'(A)') '**********************************************'
         stop
      end if
C
C  First fall is end of rise
C
      do while (j.gt.0.and.i.le.n)
         if (theta(i).le.theta(i-1)) j=0
         i=i+1
      end do
      if (i.lt.n) then
         n_rise=i+15
      else
         WRITE(6,'(A)') '**********************************************'
         WRITE(6,'(A)') '* ERROR: Data does not contain a first max. *'
         WRITE(6,'(A)') '**********************************************'
         stop
      end if
      ELSE                            ! Data does not start at zero position
      n_zero=0
      n_rise=0
      i=1
      END IF
C
C  Beginning of data occurs when there are five lower readings in a row occur
C
      j=0                  ! Count of number of falls in a row
      do while (j.le.5.and.i.le.n)
         if (theta(i).lt.theta(i-1)) then
            j=j+1
         else
            j=0
         end if
         i=i+1
      end do
      if (i.lt.n) then
         n_start=i-7
      else
         WRITE(6,'(A)') '************************************'
         WRITE(6,'(A)') '* ERROR: Data has no first minimum *'
         WRITE(6,'(A)') '************************************'
         stop
      end if
C
C End of data occurs when a relative maximum is less than 1/20 of first maximum
C
      do while (theta(i).gt.0.05*theta(n_start).and.i.le.n)
         call find_next_min(theta,i,n,n_stop)
         i=n_stop+1
         call find_next_max(theta,i,n,n_stop)
         i=n_stop+1
      end do
      call find_next_min(theta,i,n,n_stop)
      i=n_stop+1
      if (i.ge.n) then
         WRITE(6,'(A)') '**********************************'
         WRITE(6,'(A)') '* ERROR: Cannot find end of data *'
         WRITE(6,'(A)') '**********************************'
         stop
      end if
C
C  Normal return
C
      return
      end



      subroutine find_next_max(theta,i,n,j)
C
C  Find the next maximum after point theta(i)
C
      implicit none
C
C  Declare variables
C
      real theta(10001)               ! Measured angular position
      integer i                       ! Starting point in array
      integer n                       ! Number of data points
      integer j                       ! Location of next maximum
      integer max_loc                 ! Latest maximum location
      real max_val                    ! Latest maximum value
C
C  Initialize search
C
      j=i
      max_loc=i
      max_val=theta(i)
C
C  Serach to next maximum
```

```
C
      do while (
     $    .not.(theta(j  ).ge.theta(j+1).and.theta(j+1).ge.theta(j+2)
     $     .and.theta(j+2).ge.theta(j+3).and.theta(j+3).ge.theta(j+4)
     $     .and.theta(j+4).ge.theta(j+5)).and.j.le.(n-5))
         j=j+1
         if (theta(j).gt.max_val) then
            max_val=theta(j)
            max_loc=j
         end if
      end do
      j=max_loc
C
C  Normal return
C
      return
      end


      subroutine find_next_min(theta,i,n,j)
C
C  Find the next minimum after point theta(i)
C
      implicit none
C
C  Declare variables
C
      real theta(10001)                ! Measured angular position
      integer i                        ! Starting point in array
      integer n                        ! Number of data points
      integer j                        ! Location of next maximum
      integer min_loc                  ! Latest minimum location
      real min_val                     ! Latest minimum value
C
C  Initialize search
C
      j=i
      min_loc=i
      min_val=theta(i)
C
C  Serach to next minimum
C
      do while (
     $    .not.(theta(j  ).le.theta(j+1).and.theta(j+1).le.theta(j+2)
     $     .and.theta(j+2).le.theta(j+3).and.theta(j+3).le.theta(j+4)
     $     .and.theta(j+4).le.theta(j+5)).and.j.le.(n-5))
         j=j+1
         if (theta(j).lt.min_val) then
            min_val=theta(j)
            min_loc=j
         end if
      end do
      j=min_loc
C
C  Normal return
C
      return
      end
      subroutine find_zero(theta,n_zero,n_stop,zero)
C
C  Find the zero reference point
C
      implicit none
C
C  Declare variables
C
      real theta(10001)                ! Measured angular position
      integer n_zero                   ! Index for last zero measure
      integer n_stop                   ! Last value of theta
      real zero                        ! Zero reference voltage
      integer i                        ! Counter
C
C  Test if data starts at zero
C
      if (n_zero.eq.0) then  ! Average over last five theta values
         zero=0.
         do i=n_stop-5,n_stop
            zero=zero+theta(i)
         end do
      else                   ! Use resting data points
C
C  Average over the first n_zero values of theta
C
         zero=0.
         do i=1,n_zero
            zero=zero+theta(i)
         end do
         zero=zero/float(n_zero)
      end if
C
C  Normal return
C
      return
      end


      subroutine find_start(theta,n_rise,n_start,start)
C
C  Find the starting reference voltage
C
      implicit none
C
C  Declare variables
C
      real theta(10001)                ! Measured angular position
      integer n_rise                   ! Index for last rise to starting angle
      integer n_start                  ! Index for last initial anglemeasure
      real start                       ! Starting reference voltage
      integer i,k                      ! Counter
C
C  Find the average for the starting angle voltage
C
```

```
      start=0.
      k=0
      do i=n_rise+1,n_start
         start=start+theta(i)
         k=k+1
      end do
      start=start/float(n_start-n_rise)
C
C  Normal return
C
      return
      end


      subroutine
     $   make_radians(theta,initial_deg,zero,start,n)
C
C  Convert the voltages to angles in radians
C
      implicit none
C
C  Declare variables
C
      real theta(10001)                ! Measured angular position
      real initial_deg                 ! Initial angle in degrees
      integer n                        ! Number of measured points
      real zero                        ! Zero reference voltage
      real start                       ! Starting reference voltage
      real pi                          ! 3.14159265358979
      integer i                        ! Counter
C
C  Set constants
C
      pi=3.14159265358979
C
C  Convert the voltage readings to angles in radians
C
      do i=1,n
         theta(i)=(theta(i)-zero)*initial_deg/(start-zero)*pi/180.
      end do
C
C  Normal return
C
      return
      end


      subroutine interpolate_maximum(time,theta,n,theta_max,dt)
C
C  Interpolate to find relative maximum given three points
C      & zero crossing interval
C
      implicit none
C
C  Declare variables
C
```

```
      real time(10001)                ! Time at measured angle
      real theta(10001)               ! Measured angular position
      real theta_max                  ! Current maximum
      real dt                         ! Zero crossing interval
      integer n                       ! Center point in interpolation
      integer i,j,k                   ! Counter
      real*8 x(7),y(7)                ! Three point used to find extreme
      real xmax,ymax                  ! Maximum values from interpolation
      real*8 det                      ! Determinant of coefficient matrix
      real*8 a(3,3)                   ! Coefficient matrix
      real*8 b(3)                     ! Right hand side coefficients
      real*8 c(3)                     ! Quadratic coefficientsa
      real*8 dydx(7)                  ! Slopes at points
C
C  Quadratic is: y = c(3)*x**2+c(2)*x+c(1)
C
C  Set the points in the interpolation
C
      do j=1,7
         x(j)=time(n+j-4)
         y(j)=theta(n+j-4)
      end do
C
C  Find sums for coefficient matrix and right hand sides
C
      do i=1,3
         b(i)=0.
         do j=1,3
            a(i,j)=0.
            do k=1,7
               a(i,j)=a(i,j)+x(k)**(i+j-2)
            end do
         end do
         do k=1,7
            b(i)=b(i)+y(k)*x(k)**(i-1)
         end do
      end do
C
C  Solve the set of simultaneous equations by Cramer's rule
C
      det=a(1,1)*a(2,2)*a(3,3)+a(1,2)*a(2,3)*a(3,1)+a(1,3)*a(2,1)*a(3,2)
     $   -a(3,1)*a(2,2)*a(1,3)-a(3,2)*a(2,3)*a(1,1)-a(3,3)*a(2,1)*a(1,2)
      c(1)=b(1)*a(2,2)*a(3,3)+a(1,2)*a(2,3)*b(3)+a(1,3)*b(2)*a(3,2)
     $    -b(3)*a(2,2)*a(1,3)-a(3,2)*a(2,3)*b(1)-a(3,3)*b(2)*a(1,2)
      c(2)=a(1,1)*b(2)*a(3,3)+b(1)*a(2,3)*a(3,1)+a(1,3)*a(2,1)*b(3)
     $    -a(3,1)*b(2)*a(1,3)-b(3)*a(2,3)*a(1,1)-a(3,3)*a(2,1)*b(1)
      c(3)=a(1,1)*a(2,2)*b(3)+a(1,2)*b(2)*a(3,1)+b(1)*a(2,1)*a(3,2)
     $    -a(3,1)*a(2,2)*b(1)-a(3,2)*b(2)*a(1,1)-b(3)*a(2,1)*a(1,2)
      c(1)=c(1)/det
      c(2)=c(2)/det
      c(3)=c(3)/det
```

```
c
c  Find the extremes
c
      xmax=-c(2)/(2.*c(3))
      ymax=c(1)-c(2)**2/(4.*c(3))
c
c  Return the maximum
c
      theta_max=ymax
c
c  Find zero crossing interval
c
      do i=1,7
         dydx(i)=2.*c(3)*x(i)+c(2)
      end do
      if (det.eq.0.) then      ! This actually happens
         dydx(7)=(y(7)-y(6))/(x(7)-x(6))
         dydx(1)=(y(2)-y(1))/(x(2)-x(1))
      end if
      dt=x(7)-x(1)-(y(7)/dydx(7)-y(1)/dydx(1))
      if (dt.lt.0) dt=1.e-10
c
c  Normal return
c
      return
      end
```